1999

Robust optimal motion-tracking control of flexible-joint robots.

Frank. Ciuca
University of Windsor

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

ROBUST OPTIMAL MOTION-TRACKING CONTROL OF FLEXIBLE-JOINT ROBOTS

by

Frank Ciuca

A Thesis
Submitted to the College of Graduate Studies and Research
through the Program of Mechanical Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

1999

ABSTRACT

Joint flexibility exists to some degree in most industrial robots and must be taken into account in order to achieve the desired robot performance. Several flexible-joint robot dynamic models exist, and various control schemes have been developed based on these dynamic models. Most of these existing controllers are nonlinear and are complex to implement in industry. Ideally, controllers have the following properties: i) do not require full state feedback, ii) do not require noise-free measurements, and iii) are robust. However, existing nonlinear controllers for flexible-joint robots possess at most only two of these properties. Several linear controllers exist and others could be derived to possess all of the ideal-controller properties. However, because existing linear dynamic models are only accurate under limiting conditions, existing linear controllers are also limited in application.

In this thesis an acceptable and accurate linear dynamic model for flexible-joint robots was derived and, based on this model, a linear position-tracking controller for a study-case flexible-joint robot, which possesses all of the above-mentioned controller properties, was also derived.

A linear dynamic model was derived through a linearization of another, proposed, nonlinear dynamic model. This proposed nonlinear dynamic model was a more detailed version of existing nonlinear dynamic models. A form of the proposed linear dynamic model was derived for the study-case robot and simulations were carried out to demonstrate its accuracy and validity. The results of these simulations indicate that the proposed study-case linear dynamic model provides good tracking of the corresponding

nonlinear (true) dynamic model, over the entire range of operations. This implies that the proposed study-case linear model can be considered an acceptable representation of the true study-case robot dynamics. In order to allow the proposed linear model to provide an even better representation of the corresponding nonlinear dynamic model, the linearization error of the proposed linear model was modeled as a stochastic plant disturbance and included in this model.

The proposed motion-tracking robust controller was then derived for the study-case robot, based on the proposed linear dynamic model. This controller was designed using robust optimal linear LQG/LTR control techniques. This controller took into account plant disturbances and measurement noise and used only link and motor position feedback. Simulations were carried out to demonstrate the controller's performance and robustness characteristics. The results of these simulations indicate that the proposed controller provides good motion tracking and is quite robust against plant disturbances and measurement noise.

DEDICATION

to matthew, claudia, and james

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. H. ElMaraghy, as well as Dr. Lahdhiri, for their help and guidance throughout this work. Without them this thesis would not have been possible.

I would also like to thank the technicians Mr. R. Barakat and Mr. D. McKenzie, who were always available and willing to help me, even when they had a cold.

Thank you

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: INTRODUCTION

1.1 INTRODUCTION

Stiffness is a property of all metal parts, and is proportional to the modulus of elasticity of the metal, and the geometric characteristics of the part. In robot manipulators, flexibility is categorized as link flexibility and joint flexibility, depending on which system it affects. Link flexibility refers to the elasticity present in the link components and the fasteners that hold the links together. Joint flexibility refers to the flexibility present in the drive components, such as the drive shafts, the drive gears, and/or the drive cables.

Link flexibility results in a deflection of the link (Figure (1-1)) and depends on the link stiffness and length. The traditional method of counteracting link flexibility is to increase the links' cross-sectional moments of inertia. Joint flexibility results in a difference between the position dictated by the link actuator and the actual position manifested by the actuated link (Figure (1-2)). Joint flexibility is not as simple to compensate for as link flexibility since joint geometry can generally not be modified. The problem of joint flexibility is also more prominent today than in the past due to the increased use of specific drives. High-torque/low-speed drives have been used in the past because they minimize the coupling effects between the robot components and result in low backlash, and low joint flexibility. However, as these drives are heavy and bulky, in today's industry, efforts are being made to replace them with more compact configurations. These new configurations possess more joint flexibility. Harmonic

drives are an example of this problem. Harmonic drives are increasingly used in industry because they can offer the same gear ratios with much smaller gear sizes. This is achieved through the use of a non-rigid flexible spine, which, naturally, results in increased flexibility at the robot joints.

In the mid 80's several studies were performed to research the effects of joint flexibility in industrial robots; e.g. Good et al, '85 [12], Sweet and Good, '85 [30]. The studies indicated that the majority of the robots possess some degree of joint flexibility and that their joint drives could initiate resonant behavior at frequencies as low as 8 Hz [30]. Such low frequencies fall within the bandwidths of most existing controllers and are generally not taken into account in the control techniques designed for rigid robots. If not accounted for, the low resonant frequencies can be excited, which results in a degradation of the controller performance, and in some extreme cases, unstable behavior. Hence, the joint flexibility has to be accounted for in the modeling and control of robot manipulators, in order to provide precision task execution and prevent unexpected behavior.

1.2 LITERATURE SURVEY ON MODELING AND CONTROL OF FLEXIBLE JOINT ROBOTS

In order to account for joint flexibility in robotic applications, a realistic dynamic model is needed to derive and design the appropriate controller with which the desired performance is achieved. The issues of modeling and control are discussed in this section and a comprehensive literature survey is provided.

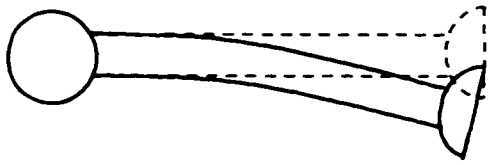


Figure 1-1: The difference between the desired position (dashed) and the achieved position (solid) in a flexible link

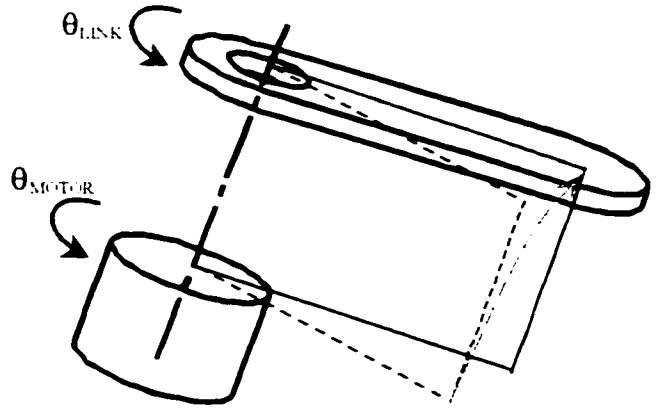


Figure 1-2: The difference between the desired position (dashed) and the achieved position (solid) in a flexible joint

1.2.1 MODELING TECHNIQUES

There are two approaches to modeling joint flexibility in robotic manipulators. The first approach is an extension of the modeling techniques used for flexible-link robots. The link-flexibility problem was considered first, and for this case, the use of equations based on the natural frequencies of vibration is still the most common modeling technique. As a result, when dealing with flexible-joint robots, the first instinct is to extend the flexible-link dynamic models. To this end, the work of Book '93 [2] presents the development of a set of transformation matrices that would allow the modeling of both flexible-link and flexible-joint robot dynamics. The transformation matrices are to be applied to the general standard 4x4 D&H matrix robot representation. The elements of the transformations matrices are computed as sums of terms based on the natural frequencies of vibration. Simulation results are presented in [2] for a flexible-beam robot and it is indicated that this procedure could be extended to the case of flexible-joint robots. However, it is unclear how this could be achieved since the natural frequencies of vibration are mostly lateral, while the vibrations present in a flexible joint are mostly angular. In the work of ElMaraghy et al '94 [10], the answer to the previously discussed problem of [2] is provided, indicating that this method could be used for specially designed robots whose vibrational torsional modes occur at frequencies well below the frequencies of other vibrational modes.

The second approach is an extension of the rigid-robot Euler-Lagrange models. The equations used in this approach contain derivatives of summations of energies. Unlike a rigid joint which is characterized by only one degree of freedom (the motor angle), a flexible joint is characterized by two degrees of freedom (the motor angle and

the link angle). To model this additional degree of freedom, a pseudo- link/joint pair is inserted at each joint and the dynamic model is derived through the same methods as the rigid-robot model. The resulting dynamic model for flexible-joint robots is similar to the dynamic model for rigid robots, with the exception of additional terms which account for the elastic energy present in the flexible joints.

In recent years, feedback linearization has emerged as the most promising nonlinear control technique for robot motion tracking. However, the dynamic model for flexible-joint robots derived with the above-mentioned approach is not feedback linearizable, because the system is not controllable; (Cesareo and Marino, '84 [3], De Simone and Nicolo '86 [6]). To make this dynamic model feedback-linearizable, in the work of Spong '87 [29] it is proposed that the following simplifying assumptions must be made for the dynamic model: (1) the robot motors are symmetric about their axis of rotation, (2) the kinetic energy of each rotor is due only to its own rotation, and (3) the joint flexibility behaves as a linear spring. Assumption (2) is the least plausible, in essence indicating that the effect of the velocities of links and rotors j, $j \neq i$, on the centrifugal and coriolis acceleration of rotor i is negligible. This assumption limits the application of the model of [29] to slow-motion robotic applications. Also, it is worth mentioning that the model of [29] consists of two second-order dynamic equations, with the position and velocities of the links and the motors as the system's states. However, the most commonly used state-space form of this model is a fourth-order model, where the states are the joint's angular positions, velocities, accelerations, and jerks; (e.g. ElMaraghy et al, '94 [10], Lahdhiri and ElMaraghy '97 [17], Massoud '94 [21]).

Another approach to making the nonlinear dynamic model feedback linearizable is through the use of the singular perturbation technique. This technique is based on the idea of decomposing the dynamic model into two sub-models based on time scale: a fast dynamics sub-model and a slow dynamics sub-model. This separation is based on the assumption that the fast dynamics of the system are much faster than the slow dynamics of the system. For flexible-joint robot dynamic models, the fast-dynamics sub-model describes the flexible-motion dynamics, and the slow-dynamics sub-model describes the rigid-motion dynamics. However, joint flexibility cannot be neglected and, therefore, the singular perturbation technique is limited to weak-joint flexibility applications. This model is used in such research works as Slotine and Hong '86 [27], and Spong, '89 [28].

Also, it is observed that in many pieces of research work, e.g. De Luca, '88 [5], Nicosia and Tomei '95 [23], Slotine and Hong '86 [27], Spong, '87 [28], the effect of friction is not included in the nonlinear dynamic model, in order to simplify the controller design. However, friction plays an important role in the model dynamics and should not be neglected (Sciavicco and Siciliano '96 [26]).

The dynamic models for flexible-joint robots discussed thus far are nonlinear. Several linear dynamic models for flexible-joint robots are possible, but a literature survey indicates that only a few researchers have used them. The simplest and most commonly used model in the industry for rigid robots considers each motor/link group as a Single-Input (the motor's torque) Single-Output (the link's motion) system. The interaction between these SISO systems is most often neglected, or is taken into account as a plant disturbance. This model can be extended for flexible-joint robots by the insertion of the additional degree of freedom at each joint. Each SISO system is now the

motor/flexible-joint/link group. As long as the flexible joint is modeled by a linear spring, the SISO system continues to be characterized by a linear dynamic model. The parameters (e.g. the motor's moment of inertia, the joint's stiffness) of such a model can be simply taken exactly as the robot parameters, or can be determined from a sine sweep performed for the entire system. Both methods have their limitations. For the first method, the error is due to the summation/averaging of the properties of several components in one parameter. For example, the one value for link viscous friction actually refers to the average viscous friction of all the components that are part of that "link". For the second method, the error is due to the need to assume an order for the transfer function characterizing the system before the sine sweep can be carried out. A SISO system model with the parameters identified by the first method outlined above is most often used by research performed on single-link flexible-joint robot case studies. In the work of Massoud '94 [21] a linear dynamic model is developed for an experimental two-link flexible-joint robot, a third-order transfer function is assumed for this model, and the system parameters are identified by performing a sine sweep. In general, these SISO flexible joint robot models are only accurate at low speeds, where the interaction between the joint dynamics is negligible.

Another method of deriving the linear model is through Taylor series expansion of the nonlinear model. Unlike the SISO linear models, the linearized nonlinear models still contain some of the terms that describe the interaction between the individual motor/flexible-joint/link sub-systems. As a result, these models are more complex, but are also expected to provide the best tracking of the nonlinear (true) robot behavior of all linear models. No use of this model is found in the literature. These linear models are

most accurate for operations near the linearization point, and become less accurate as the system's states move away from this linearization point.

The third linear dynamic model used in the studies of flexible-joint robots is the same linear dynamic model as that for rigid-robots. The use of this approach is justified since the mathematical *passivity property* of the rigid-robots still holds for flexible-joint robots. This approach is used by Tomei '91 [31] and Tomei '94 [33], but no discussion of the accuracy is presented by either of these. Since it has been shown in the work of Good et al '85 [12], and Sweet and Good, '85 [30], that there is a large discrepancy between the behavior exhibited by flexible-joint drives and the behavior dictated by rigid-robot models, any rigid-robot models is expected to provide poor representation of flexible-joint robot models.

The first objective of this thesis is to derive an acceptable comprehensive dynamic model for the flexible-joint robot manipulator under study. The second objective of this thesis is to design a robust controller based on the proposed dynamic model. A literature survey of control techniques for flexible-joint robots is discussed next.

1.2.2 CONTROL TECHNIQUES

Various control algorithms are derived in order to achieve the specified system characteristics. The effectiveness of these designed controllers is then evaluated based on several factors, of which robust stability and robust performance are the most important. Stability is an indication of whether the system will reach a stable steady-state value, or will become unstable and crash. Performance refers to how well the system tracks any arbitrary reference trajectory. Robustness refers to how resilient the system is towards

system parameters uncertainty and unexpected disturbances. In the course of the control action, it is necessary to measure and feedback the resulting system output, but this is not always possible in industry. Some quantities may not be easily measurable and/or it may be very costly to do so. Hence, in addition to having good performance and robustness, it is also desirable that the controller minimize the number of feedbacks used.

The commonly-used control techniques for flexible-joint robots fall in two distinct classes: linear and nonlinear. Nonlinear methods are discussed first.

Nonlinear methods can further be subdivided into those two groups: inverse dynamics / feedback linearization -based controllers and Lyapunov -based controllers. In inverse dynamics / nonlinear feedback control, a feedback loop is used to cancel out the plant nonlinearities, resulting in a linear model. Spong presented nonlinear feedback control based on the fourth-order state-space model in Spong '87 [29], and later developed a control technique for the slow dynamics in the singular-perturbation -based model of Spong '89 [28]. In the work of Slotine and Hong '86 [27], the nonlinear feedback method of sliding controller is used to control the fast variables of the system, assuring that the overall model dynamics fall within the slow manifold ($\approx$ variable subspace) required by the control approach of [28]. In the work of De Luca '88 [5], a controller is developed based on a dynamic model which does not require assumptions (2) of [29], i.e. the kinetic energy of each rotor is due only to its own rotation. This is accomplished through the use of *dynamic* nonlinear feedback. However this approach imposed limitations on the desired robot trajectory [5].

There are several limitations with the control methods discussed thus far and listed above. Beside the limitations on the dynamic modeling, the control techniques often require the feedback of all states, exact knowledge of robot parameters, and noise-free measurements. The requirement of full-state feedback is particularly difficult/expensive for the controller designs requiring measurement of acceleration and jerk. Exact knowledge of parameters is also not usually feasible. In industry, only approximate values for these robot parameters are determined through finite analysis methods or through experimental testing. Finally, all sensor equipment is affected by digitalization error and can further be corrupted by various sources, resulting in measurements with some degree of noise.

Several researchers, such as Nicosia and Tomei '95 [23], Nicosia et al '88 [24], and Tomei, '90 [32], offered various observers to reduce the number of sensors required, but the design of these observers is highly nonlinear, which further complicates the controller design. In addition, a common challenge to all these design techniques is the problem of robustness towards model uncertainties and system parameters variations.

Other researchers, e.g. Ghorbel and Spong '92 [11], Jankowski et al '93 [15], Lammerts et al '95 [18], Massoud '94 [21], proposed inverse dynamics -based and adaptive controllers based on Lyapunov methods to improve the controller performance. Adaptive controllers are designed such that their gains change (adapt) during the course of the control action, resulting in improved performance and robustness. In the work of Jankowski et al '93 [15] the technique of inverse dynamics is used, and a controller is developed whereby robustness is achieved through robust servomechanism theory and through Lyapunov's second method. In the work of Qu '95 [25] a robust control method

is developed, but requires extremely high control torques. Two robust adaptive controllers are also developed in the work of Massoud '94 [21]. However, most of these robust controllers developed, e.g. Jankowski et al '93 [15], Lin and Yu '96 [19], Massoud '94 [21], are only robust towards uncertainty in the robot parameters, and not towards unexpected plant disturbances. Furthermore, the parametric uncertainty tested in the simulations presented in these papers is always of constant magnitude, e.g. exactly +0.98kg to the end effector mass [21] or exactly 10% in arm inertias [19]. Furthermore, the majority of these robust controllers required full-state feedback, and assumed noise-free measurements. In the work of Lahdhiri and ElMaraghy '97 [17] a controller is derived which does not require full state feedback and takes into account measurement noise, but did not take into account model uncertainties. The controller in [17] is derived through the application of nonlinear feedback theory, coupled with linear robust methods.

Since accurate linear dynamic models for flexible joint robots do not exist in the literature, far fewer linear controllers have been proposed. In the work of Massoud '94 [21] a PD controller is derived for the [21] linear model discussed in Section 1.2.1 and implemented on an experimental two-link flexible-joint robot. The results of this implementation indicated that the PD controller provides unsatisfactory tracking performance, in that, i) the overshoot is too high, ii) the response time is too low, and iii) the settling time is too long. A PD controller is also developed in Tomei '91 [31], for the [31] linear model discussed in Section 1.2.1, and tested on a three-link flexible joint robot simulation. The results of [31] indicate that the PD controller provided asymptotic tracking of the reference signal when there is no parametric uncertainty, and tracking with a finite steady-state error when there is parametric uncertainty. In the work of Chen

et al '1998 [4] a PID controller and a fuzzy PID controller are developed and tested on an experimental setup of a single-link flexible- joint arm. The [4] results indicate that the PID controller can not provide satisfactory reference signal tracking, while the fuzzy PID controller provides much better tracking, with a steady-state error of 1.6%. In the work of Lin and Yu '96 [19] a PD robust controller is developed through the use of Lyapunav functions, and the experimental results of the controller implementation are satisfactory. However, the model used in [19] is a simplified nonlinear dynamic model, and the controller is tested only on a single-link flexible-joint manipulator.

An indication of the status of control of flexible structures in industry and space technology is given in Whittaker et al '91 [34], where it is stated that "the technique for space manipulators was, and continues to be, to move the joints slowly and wait for the tip of the arm to settle to equilibrium".

1.1.3 LITERATURE SURVEY SUMMARY

In summary, most of existing nonlinear control techniques in the literature are highly complex, are often not robust, often require full state feedback, and often require noise-free measurements. Furthermore, these control techniques impose additional requirements on the dynamic model and simplifying assumptions are necessary.

On the other side, linear control techniques can take into account more details but are limited to operations near the linearization point or at very slow speeds. The linear dynamic model provides a good representation of the nonlinear dynamics only in these conditions. This modeling limitation has perhaps discouraged researchers to design

linear controllers and the literature survey indicates that fewer studies were conducted in this area.

1.3 MOTIVATION AND OBJECTIVE

Motivated by these limitations, the objective of this thesis is to design a linear robust position-tracking controller for flexible-joint robots which takes into account plant uncertainties, plant unmodelled dynamics and measurement noise. In order to achieve the desired performance and stability, a comprehensive and realistic model is derived first.

This model has the advantages that it takes into account the effects of friction, coriolis and centrifugal acceleration, and nonlinear elastic element characteristics. Stick-slip friction models are used in this work and replace the classical coulomb and static friction model. This is due the fact that recent studies, e.g. Dupont '93 [9], Haessig and Friedland '91 [13], indicate that stick-slip friction curves model more accurately friction behavior. Nonlinear elastic elements are used to replace the linear elastic elements since they are more likely to provide a more correct representation of joint flexibility in industrial robots than linear elastic elements. A linear model is then derived through a truncated Taylor series expansion of the nonlinear model. The truncation error is taken into account and is modeled as a plant disturbance acting on the input. This disturbance is considered as Gaussian noise, and its statistical characteristics is determined using extensive simulations. The measurement noise is considered a plant disturbance acting on the output, and is modeled as a Gaussian noise whose statistical characteristics are dictated by the sensor specifications.

In order to validate the proposed model, extensive simulations are performed in MATLAB™[22] for a study case consisting of an experimental, two-link, planar, revolute, flexible joint robot, shown in Figure (1-3). This robot was designed and built at the Flexible Manufacturing Centre (ElMaraghy et al '94 [10], Massoud '94 [21]), where it was used for several pieces of research work sponsored by the Institute for Robotics and Intelligent Systems (IRIS).

The second part of this thesis consists of designing a robust linear position-tracking controller for the study-case flexible joint robot. This controller design is based on the proposed linear dynamic model and is derived using Linear Quadratic Gaussian (LQG) / Loop Transfer Recover (LTR) techniques. In order to reduce the cost of the proposed controller, the links' and motors' positions are the only measurements used. Extensive simulations are carried out to demonstrate the position tracking ability, stability and performance of this controller. The controller design and simulations are carried out using MATLAB™ .

The third and final objective of this thesis is to prepare this controller for an experimental implementation on the available study-case robot. With this scope, the proposed controller is described in implementation-ready algorithmic form. Future work will deal with the physical implementation of this algorithm.

1.4 THESIS OVERVIEW

This dissertation of this thesis is organized as follows. Chapter 2 presents the derivation of the dynamic model for flexible joint robots. This chapter includes the

derivation of the nonlinear detailed model, the derivation of the linear detailed model, and the model validation. Chapter 3 contains the derivation of the control algorithm. Chapter 4 contains the details of the experimental setup and the experimental controller algorithms. Chapter 5 contains some conclusions and recommendations for future work. Appendix 1 contains the details of the derivation of the linear dynamic model written explicitly for the study-case robot.

On a final note on the text of this dissertation, it is indicated that variables in bold format refer to matrix or vector quantities, whereas variables not in bold format refer to scalar quantities. Another idiosyncrasy, typical of the study of robot dynamics or control and used in this dissertation, is the use of a dot between the brackets of a function statement (e.g. X(.)), to refer to "all the dependent variables" of that function. The use of the "." is necessary when the function in question may have different dependent variables depending on its application, and it is a shorthand if the dependent variables of the function have already been indicated in a previous statement. When used with a matrix, the "." can denote an entire row vector (e.g. M (2 , .) = second row vector of matrix M) or an entire column vector (e.g. M (. , 3) = third column vector of matrix M).

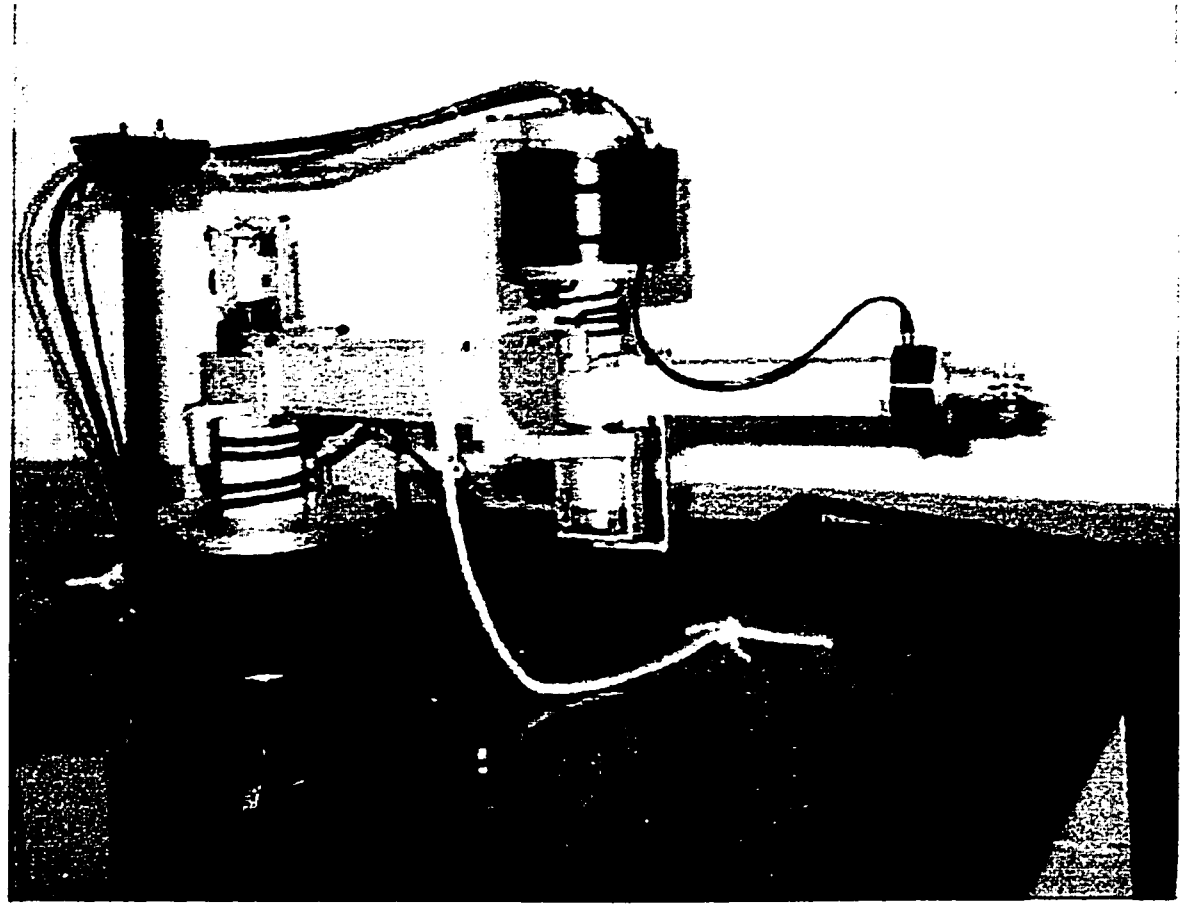


Figure 1-3: Experimental Flexible-Joint Robot
(ElMaraghy et al '94 [10], Massoud '94 [21]).

Chapter 2: DYNAMIC MODEL

2.1 INTRODUCTION

As discussed in Chapter 1, nonlinear dynamic models impose several restrictions on the control approach used. Linear models impose less restrictions on the controller design. However, existing linear models for flexible-joint robots are only valid at low speeds or for operating conditions where the system states are within a close neighborhood of the linearization point. The objective of this chapter is to develop a realistic and comprehensive linear model which overcomes these limitations of existing linear models.

The proposed linear dynamic model is obtained through the linearization of a proposed nonlinear dynamic model. The proposed nonlinear dynamic model is an extension of an existing nonlinear dynamic models for flexible-joint robots (Massoud '94 [21]), with additional terms to better model the joint flexibility and the friction. The proposed linear dynamic model is compared to its corresponding nonlinear dynamic model in a series of simulations for the study-case robot. The simulations results indicate that the linear model provide accurate tracking of its corresponding nonlinear dynamic model over the whole range of possible trajectories. The simulations results are also used to derive a stochastic model of the error between the proposed linear model and the nonlinear model. This error term is then included in the linear dynamic model. This linear dynamic model with uncertainty terms is used to derive a robust controller for the study-case flexible-joint robot in Chapter 3.

This chapter is organized as follows. Section 2.2 contains the derivation of the proposed dynamic linear model for the general n-link flexible-joint robot, and for the study-case two-link flexible-joint robot. The optimization and validation of the proposed model is described next, in Section 2.3. Section 2.4 contains the final summarized linear dynamic model for the study-case robot, which is used in the controller design in Chapter 3. Section 2.5 contains some conclusions.

2.2 DYNAMIC MODELING

Since existing dynamic models for flexible-joint robots are an extension of the dynamic model for rigid robots, this section starts with an overview description of the n-link rigid-robot model. The most detailed n-link nonlinear dynamic model for flexible-joint robots available in the literature is then presented, followed by the proposed n-link nonlinear and linear dynamic models for flexible-joint robots, in Sections 2.2.2 and 2.2.3, respectively. Section 2.2.4 describes the model chosen for the plant disturbance. Section 2.2.5 contains some details regarding the proposed linear dynamic model for the study case robot.

2.2.1 INTRODUCTION

The Euler-Lagrange dynamic model for a rigid robot manipulator is given by:

$$\mathbf{D(q)\ddot{q} + C(q,\dot{q}) + B\dot{q} + Fc = U - J^T F} \qquad (2.1)$$

In Equation (2-1), $\mathbf{q}$ is the vector of joint variables (e.g., $q_i = \theta_i$ for revolute joint i, $q_i = d_i$ for prismatic joint i), the $\mathbf{D(q)}$ term accounts for the robot inertia, the $\mathbf{C(q,\dot{q})}$ term accounts for the centrifugal and coriolis acceleration, the $\mathbf{B\dot{q}}$ term accounts for the

viscous friction, the $\mathbf{F_C}$ term accounts for the coulomb/static friction, $\mathbf{U}$ denotes the vector of system input forces, and the $\mathbf{J^T F}$ term denotes forces resulting from interaction with the environment (e.g. from contact). This model is derived through summation of energies and the application of the Lagrange equation. Expression for the terms **D(.)** and **C(.)** for any robot configurations are available in most robot textbooks, e.g. Sciavicco and Siciliano '96 [26].

A literature survey on robot modeling indicates that, in order to take into account the effect of joint flexibility, an additional degree of freedom is required at each joint. In order to model this additional degree of freedom in the standard robot convention, a pseudo- joint/link pair is inserted at each joint. The dynamic equations are then derived in a procedure similar to that for the robot model, resulting in an equation similar to Equation (2-1), with twice as many states. It has become standard convention for flexible-joint robot models to separate the dynamics into two sub-models: one describing the links' dynamics and one describing the motors' dynamics. The most detailed nonlinear model for an n-link flexible-joint robot, in the absence of contact forces, available in the literature, is given by Massoud '94 [21]:

$$\mathbf{D(q)\ddot{q} + \Lambda\ddot{q}_M + C(q,\dot{q}) + B_L\dot{q} + F_{CL} - K \times (q_M - q) = 0} \qquad (2\text{-}2)$$

$$\mathbf{\Lambda\ddot{q} + I_M\ddot{q}_M + B_M\dot{q}_M + F_{CM} + K \times (q_M - q) = U} \qquad (2\text{-}3)$$

where vector $\mathbf{q}$ is the same as in Equation (2-1), namely the link angular position vector, and $\mathbf{q_M}$ is the new degree of freedom, the motor angular position vector. Terms **D(.)** and **C(.)** and **U** are also the same as those in Equation (2.1). There is now viscous friction which affects both the joints and the motors, terms $\mathbf{B_L\dot{q}}$, and $\mathbf{B_M\dot{q}_M}$ respectively, and coulomb/static friction for both the joints and the motors, namely terms $\mathbf{F_{CM}}$ and $\mathbf{F_{CL}}$

respectively. The term $\mathbf{K}$ is a diagonal matrix whose elements are the joint stiffness, and the terms $\mathbf{K} \times (\mathbf{q_M} - \mathbf{q})$ account for the energy of elasticity present in the flexible joints. Finally, term $\mathbf{I_M}$ is a diagonal matrix whose elements are the moments of inertia of the motors, and the Λ term accounts for the effects of the velocities of the links and joints $j, j \neq i$, on the kinetic energy of rotor i.

2.2.2 PROPOSED N-LINK NONLINEAR DYNAMIC MODEL

In the model (2-2)-(2-3), the potential energy due to the joints' flexibility is assumed to be a linear function of the system states and represented by the term $\mathbf{K} \times (\mathbf{q_M} - \mathbf{q})$. This assumption is not generally true because of the inherent nonlinear behavior of the elasticity of industrial robots. Therefore nonlinear functions must be used, i.e. the term $\mathbf{K} \times (\mathbf{q_M} - \mathbf{q})$ must be replaced by a nonlinear function $\mathbf{K_F}(\mathbf{q_M}, \mathbf{q})$. Such a function depends on the robot system in use. There is no general analytical expression for such a function, one should be determine experimentally.

The coulomb/static friction terms $\mathbf{F_{CM}}$ and $\mathbf{F_{CL}}$ included in the model given by (2-2)-(2-3) are represented by the classical force-velocity curve shown in Figure (2-1). However, recent studies, e.g. De Wit et al '91 [7], Dupont '93 [9], Haessig and Friedland '91 [13], have indicated that the coulomb/static friction is better represented by the force-velocity curve indicated in Figure (2-2). This model is known in the literature as the "stick-slip" model.

A literature survey on friction modeling indicates that there are two approaches to including the effects of stick-slip friction in a dynamic model. The first approach (e.g.

De Wit et al '91 [7]) is based on finding a function that models the curve indicated in Figure (2-2), while the second approach (e.g. Haessig and Friendland '91 [13]) consists of modifying the control module by adding an external loop to represent the effect of friction. The first approach is more suitable when dealing with flexible-joints because it requires only the identification of the friction curve shown in Figure (2-2), while the second approach requires adjusting the compensated system after designing the controller. The most appropriate function that matches the curve of Figure (2-2) is given by De Wit et al '91 [7]:

$$F_{\text{stick-slip}}(\dot{q}) = [\varphi_0 + \varphi_1|\dot{q}|^{0.5} + \varphi_2|\dot{q}|]\,\text{sgn}(\dot{q}), \tag{2-4}$$

where the φ terms represent parameters to be identified experimentally. The function given by (2-4) has the advantage of being linear in its parameters φ_0, φ_1, and φ_2. This linear form simplifies the on-line parameter identification procedure. In the work of De Wit et al '91 [7], the slip-stick friction model (2-4) is shown to provide a close match with the observed values, as shown in Figure (2-3).

In view of this discussion, the most detailed nonlinear dynamic model for flexible-joint robot systems, which overcomes the limitations of the model (2-2)- (2-3), is given by:

$$\mathbf{D(q)\ddot{q} + \Lambda\ddot{q}_M + C(q,\dot{q}) + B_L\dot{q} + F_{CL}(\dot{q}) - K_F(q_M, q) = 0} \tag{2-5}$$

$$\mathbf{\Lambda\ddot{q} + I_M\ddot{q}_M + B_M\dot{q}_M + F_{CM}(\dot{q}_M) + K_F(q_M, q) = U} \tag{2-6}$$

Equations (2-5) and (2-6) can be put in state space form as

$$\mathbf{\dot{X} = f(X) + g(X)U} \tag{2-7}$$

where the state vector $\mathbf{X}$ is defined as $\mathbf{X} = [x_1 \quad x_2 \quad x_3 \quad x_4]^T = [q_M \quad \dot{q}_M \quad q \quad \dot{q}]^T$,

U is the control input vector consisting of the motor torques, and the functions **f(.)** and **g(.)** are given by:

$$\mathbf{f(X)} = \begin{bmatrix} \mathbf{f_1(X)} \\ \mathbf{f_2(X)} \\ \mathbf{f_3(X)} \\ \mathbf{f_4(X)} \end{bmatrix}, \text{ where} \tag{2-8}$$

$$\mathbf{f_1(x) = x_2} \tag{2-9}$$

$$\begin{aligned} \mathbf{f_2(x)} = \alpha_1(\mathbf{x_3}) \times \{ & -\mathbf{F_{CM}(x_2)} + \mathbf{k_F(x_1, x_3)} - \mathbf{B_M} \times \dot{\mathbf{q}}_\mathbf{M} + \Lambda^{\mathbf{T}} \times \\ & \times [\mathbf{D(x_3)}]^{-1} [\mathbf{B_L} \times \dot{\mathbf{q}} + \mathbf{C(x_3, x_4)} + \mathbf{k_F(x_1, x_3)} + \mathbf{F_{CL}(x_4)}] \} \end{aligned} \tag{2-10}$$

$$\mathbf{f_3(x) = x_4} \tag{2-11}$$

$$\begin{aligned} \mathbf{f_4(x)} = \alpha_2(\mathbf{x_3}) \times \{ & -\mathbf{F_{CL}(x_4)} + \mathbf{k_F(x_1, x_3)} + \mathbf{B_L} \times \dot{\mathbf{q}} + \\ & + \Lambda \times [\mathbf{I_M}]^{-1} \times [\mathbf{k_F(x_1, x_3)} - \mathbf{B_M} \times \dot{\mathbf{q}}_\mathbf{M} - \mathbf{F_{CM}(x_2)}] \}, \end{aligned} \tag{2-12}$$

and

$$\mathbf{g(X)} = \begin{bmatrix} 0 \\ \alpha_1(\mathbf{x_3}) \\ 0 \\ \alpha_2(\mathbf{x_3}) \times \Lambda \times [\mathbf{I_M}]^{-1} \end{bmatrix} \tag{2-13}$$

Terms α_1 (.) and α_2 (.) in Equations (2-10)-(2-13) are given by

$$\alpha_1(\mathbf{x_3}) = \{\mathbf{I_M} - \Lambda^{\mathbf{T}} \times [\mathbf{D(x_3)}]^{-1} \times \Lambda\}^{-1} \tag{2-14}$$

$$\alpha_2(\mathbf{x_3}) = -\{\mathbf{D(x_3)} - \Lambda^{\mathbf{T}} \times [\mathbf{I_M}]^{-1} \times \Lambda\}^{-1} \tag{2-15}$$

This concludes the derivation of the proposed n-link *nonlinear* dynamic model for flexible-joint robots. The next section describes the derivation of the proposed n-link *linear* dynamic model for flexible-joint robots.

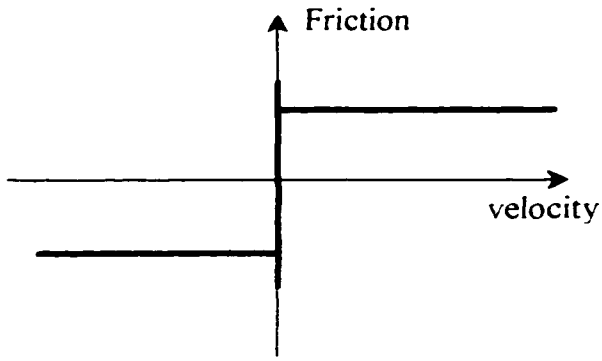


Figure 2-1: Classical Static/Coulomb Friction Model

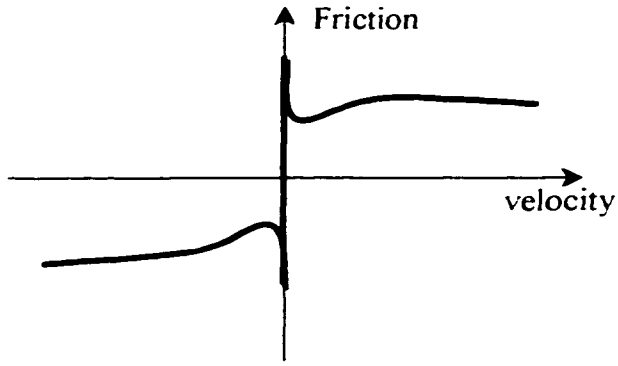


Figure 2-2: More Exact Static/Coulomb Friction Model: The Stick-Slip Model

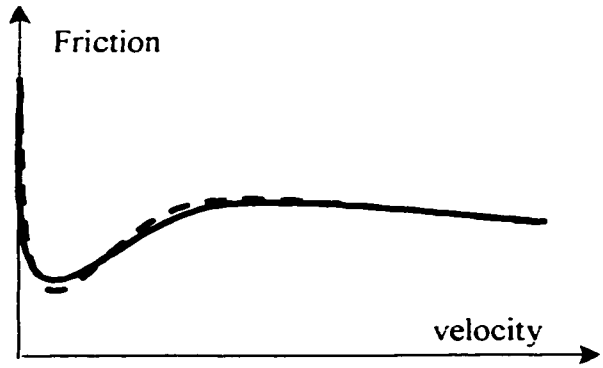


Figure 2-3: Stick-Slip Friction: observed (solid), and modeled by (2-4) (dashed)

2.2.3 PROPOSED N-LINK LINEAR DYNAMIC MODEL

In Chapter 1, several methods for deriving linear dynamic models for flexible-joint robots were discussed and it was indicated that the most accurate model is expected to be achieved through linearization of the most complete nonlinear model. The proposed linear dynamic model is obtained by linearizing the proposed detailed nonlinear model (2-7)-(2-15) using Taylor series expansion.

In the linearization of the classical nonlinear model (2-2)-(2-3) there is no real optimization process - best results are always obtained if the position variables of the optimization point are given values near the center of the region where motion is most likely to occur, and if the velocity variables of the optimization point are given 0 values, which is best for motion in both directions. However, these optimization variables are not sufficient and resulting linear models only track the nonlinear dynamic model at operating conditions near these optimization points. While the statements regarding the choices for position and velocity variables still hold for the proposed nonlinear model, this model has two more parameters per robot joint that can be varied during its linearization process: the derivatives for the coulomb/static (stick-slip) friction (i) at the motors and (ii) at the links. In the classical nonlinear model, these friction functions were of constant magnitude (refer to Figure (2-1)) and their derivatives were 0. In the proposed nonlinear model the stick-slip functions are not of constant magnitude and their derivatives with respect to velocity differ greatly for near-zero velocity values. These additional linearization/optimization parameters constitute one of the major advantages of the proposed modeling approach.

Further details regarding the optimization process are available in Section 2.3.2.

Applying Taylor series expansion to the set of Equations (2-7)-(2-15) for an n-link flexible-joint robot manipulator yields the proposed linearized system model:

$$\dot{\mathbf{X}} = \mathbf{A}\,\mathbf{X} + \mathbf{B}\,\mathbf{U} + \mathbf{G}\,\mathbf{W} \tag{2-16}$$

where the state vector **X** and the control input **U** are the same as for the n-link nonlinear model (2-7)-(2-15), term **GW** represents the error due the truncation and matrices **A** and **B** are given by:

$$\mathbf{A} = \left[\begin{array}{cccc} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_n} \\ \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2} & & \\ \vdots & & \ddots & \\ \frac{\partial \mathbf{f}_n}{\partial \mathbf{x}_1} & & & \frac{\partial \mathbf{f}_n}{\partial \mathbf{x}_n} \end{array}\right]_{\text{at } (\mathbf{X0},\mathbf{U0})} \qquad \mathbf{B} = \left[\begin{array}{c} \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}} \\ \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}} \\ \vdots \\ \frac{\partial \mathbf{f}_n}{\partial \mathbf{U}} \end{array}\right]_{\text{at } (\mathbf{X0},\mathbf{U0})} \tag{2-17}$$

2.2.4 PLANT DISTURBANCE MODELING

In Equation (2-16), **GW** describes the linearization error due to the truncation of the Taylor series, namely the difference between the actual nonlinear model and the linearized model. In other words, the **GW** term describes the unmodelled dynamics in the linear model (2-16)-(2-17), relative to the nonlinear dynamic model (2-7)-(2-15). **GW** cannot be exactly quantified by theoretical means and cannot be available online. It is at best modeled as a random error signal. It is well known that without loss of generality, any random error signal can be modeled as a Gaussian white noise [16]. However, in Equation (2-16), the **GW** term accounts for the error due to the truncation of all derivatives in Equation (2-17), and each of these derivatives depends differently on

linearization error. Therefore, in order to accurately account for this, in this thesis it was proposed that vector **W** be taken as the Gaussian white noise, and matrix **G** be taken as a weighting matrix. The entries of matrix **G** are only 0's and 1's according to the dependencies of the derivatives of included in the matrix **A** on the linearization error, multiplied by a magnitude factor *c*. Therefore, the expression of **G** is given by

$$\mathbf{G} = c \bullet \begin{bmatrix} \mathbf{g}_{11} & \mathbf{g}_{12} & \cdots & \mathbf{g}_{1n} \\ \mathbf{g}_{21} & \mathbf{g}_{22} & & \\ \vdots & & \ddots & \\ \mathbf{g}_{n1} & & & \mathbf{g}_{nn} \end{bmatrix} \tag{2-18}$$

where $\mathbf{g}_{ij}$, $i,j = 1..n$, are block matrices of size nxn, given by:

$$\mathbf{g}_{ij} = [\mathbf{all\ 1}] \text{ if } \frac{\partial \mathbf{f_i}}{\partial \mathbf{x_j}} \text{ depends on the derivatives of } \mathbf{C(.)} \text{ or } \mathbf{D(.)} \tag{2-19}$$

$$\mathbf{g}_{ij} = [\mathbf{1}] \text{ if } \frac{\partial \mathbf{f_i}}{\partial \mathbf{x_j}} \text{ does not depends on the derivatives of } \mathbf{C(.)} \text{ or } \mathbf{D(.)} \tag{2-20}$$

$$\mathbf{g}_{ij} = [0] \text{ if } \frac{\partial \mathbf{f_i}}{\partial \mathbf{x_j}} = \text{Constant,} \tag{2-21}$$

and *c* is the scalar magnitude factor reflecting the impact of the error between the linear and the nonlinear system dynamics. Several simulations are used to compute this magnitude factor, which is taken as the upper bound of the RMS of the difference between the two systems. This term is computed using the following equation:

$$c = \max\left(\sqrt{\frac{1}{n} \sum_{k=1}^{k=kend} [x_{j.nonlinear}(k) - x_{j.linear}(k)]^2}, j = 1..n \right) \tag{2-22}$$

where $x_{j.nonlinear}$ and $x_{j.linear}$ are the states of the nonlinear and the linear dynamic models obtained in time-response simulations, and kend = the number of time steps used in these simulations.

2.2.5 PROPOSED LINEAR MODEL FOR THE STUDY-CASE ROBOT

The study case robot, introduced in Chapter 1, is a two-link planar revolute flexible-joint robot. As such , the state **X** of the dynamic model for this robot is given by

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 \end{bmatrix}^T, \tag{2-23}$$

where,

$$\mathbf{x}_1 = \begin{bmatrix} x_1 = \text{motor 1 position} \\ x_2 = \text{motor 2 position} \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} x_3 = \text{motor 1 velocity} \\ x_4 = \text{motor 2 velocity} \end{bmatrix}, \tag{2-24}$$

$$\mathbf{x}_3 = \begin{bmatrix} x_5 = \text{link 1 position} \\ x_6 = \text{link 2 position} \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} x_7 = \text{link 1 velocity} \\ x_8 = \text{link 2 velocity} \end{bmatrix} \tag{2-25}$$

Because of the robot construction and the reference frame used, the motor 1 and link 1 position variables have an operating range of 0° to 180°, while the motor 2 and link 2 position variables have an operating range of –90° to 90°. Furthermore, a safe maximum operating speed on both links is considered to be 25°/s, where "s" is the abbreviation for "second".

Explicit expressions for the terms **A** and **B** of Equation (2-17) and **G** of Equation (2-18) for the study-case robot are derived in Appendix 1. Appendix 1 also contains the study-case robot parameters, and the functions chosen to represent the spring nonlinear force-displacement curves, and the stick-slip functions. The optimum linearization point is found through extensive simulations. A description of these simulations along with some details of the optimization process is presented in Section 2.3. The resulting numerical values for terms **A** and **B** of Equation (2-17) and of **G** of Equation (2-18) are presented in Section 2.4.

2.3 MODEL OPTIMIZATION AND VALIDATION

Simulations are required to compare the study-case robot linear dynamic model with its corresponding comprehensive nonlinear dynamic model. The comparison is necessary in order to optimize the linear dynamic model and then validate the optimized linear dynamic model versus the nonlinear dynamic model.

The simulations are described in Section 2.3.1. In Section 2.3.2 the optimization process of the linear dynamic model is discussed. Comparisons between the proposed linear model and previous linear dynamic models for flexible-joint robots are also presented. Section 2.3.3 contains the results of several simulations which compare the proposed optimal linear model versus the corresponding nonlinear model. This constitutes the validation of the proposed linear model. Section 2.3.4 contains the derivation of the plant disturbance term c through the use of Equation (2-22).

2.3.1 SIMULATIONS DESCRIPTION

The simulations consist of time responses of the system driven only by its initial conditions. The time response of the linear system is developed using the MATLAB^{TM} function LSIM. The time response of the nonlinear system is generated through a modified 4^{th} order Runge-Kutta integration routine.

The Runge-Kutta routine is modified in order to accurately simulate the nonlinear model which included the effects of the coulomb/stick-slip friction. Because both coulomb and stick-slip friction models dictate friction values which do not decrease as the velocity approaches zero (refer to Figure (2-1) and Figure (2-2)), the unmodified

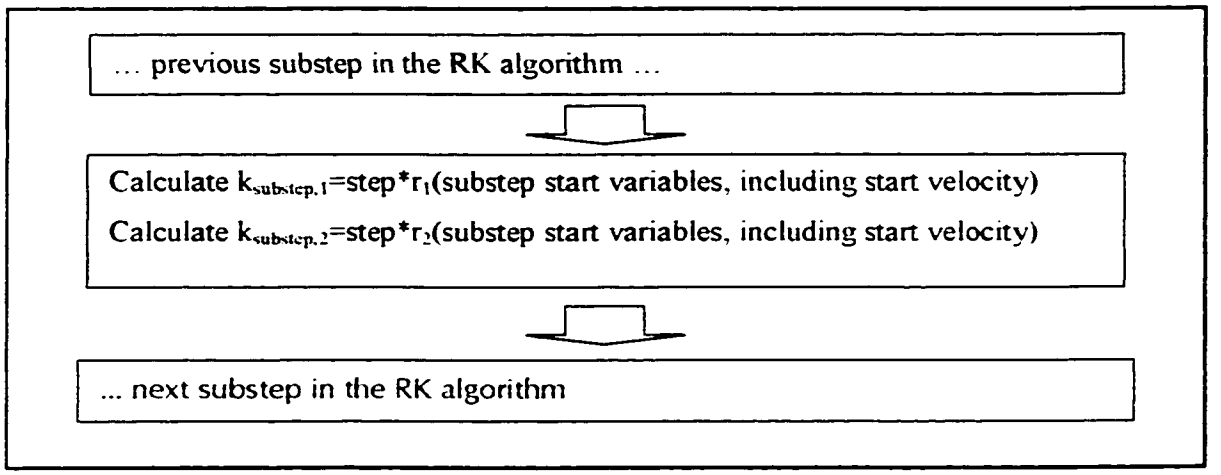
Runge-Kutta algorithm leads to changes in the velocity sign due only to the friction. This behavior is erroneous since the influence of friction can, at maximum, bring the velocity to zero. Studies on friction (e.g. Dupont '93 [9]) suggest that *variable-step* Runge-Kutta routines can be implemented to solve this problem. However, this solution is only suited for solving one differential equation and cannot be implemented in the dynamic model for n-links flexible-joint robotic systems where 4xn simultaneous equations must be solved.

In order to solve this problem, a compensation routine is proposed. This compensation routine is to be inserted within each sub-step of the Runge-Kutta routine. Beside the effects of the coulomb and static friction, the effects of the viscous friction are also removed from the regular Runge-Kutta algorithm and inserted in the compensation function. This action improves the accuracy of the friction calculation because it allows for the calculation of the viscous friction at an average sub-step velocity rather than at the sub-step start velocity. The proposed modification to the Runge-Kutta algorithm is explained in the following three figures for an example case of two ordinary differential equations $r_1(.)$ and $r_2(.)$, where $r_2(.)$ includes the effect of friction along with other terms. Figure (2-4) indicates how each Runge-Kutta sub-step is modified to accommodate the presented compensation function, while Figure (2-5) contains the schematic flowchart for the compensation function. Figure (2-6) is introduced to explain the flowchart of Figure (2-5) in detail and it is suggested that Figures (2-5) and (2-6) be read in parallel. Figure (2-6) also explains the abbreviations used in Figure (2-5).

Figures (2-7) and (2-8) contain two sample simulation results with and without the proposed modified Runge-Kutta algorithm. The simulation results obtained with the

unmodified Runge-Kutta routine show very high velocity oscillations that do not settle. This is a result of improper friction modeling. The simulation obtained with the modified Runge-Kutta routine accurately models the effects of the friction leading to a velocity time response that does not contain unrealistic high velocity oscillations. Figure (2-7) indicates that the results obtained with the proposed modified Runge-Kutta routine do retain the true velocity behavior (namely the two high-amplitude velocity oscillations) as those results obtained with the unmodified Runge-Kutta routine. This indicates that the proposed modified Runge-Kutta routine dampens out the erroneously simulated velocity oscillations without altering the true results.

(a) unmodified



(b) with proposed modification

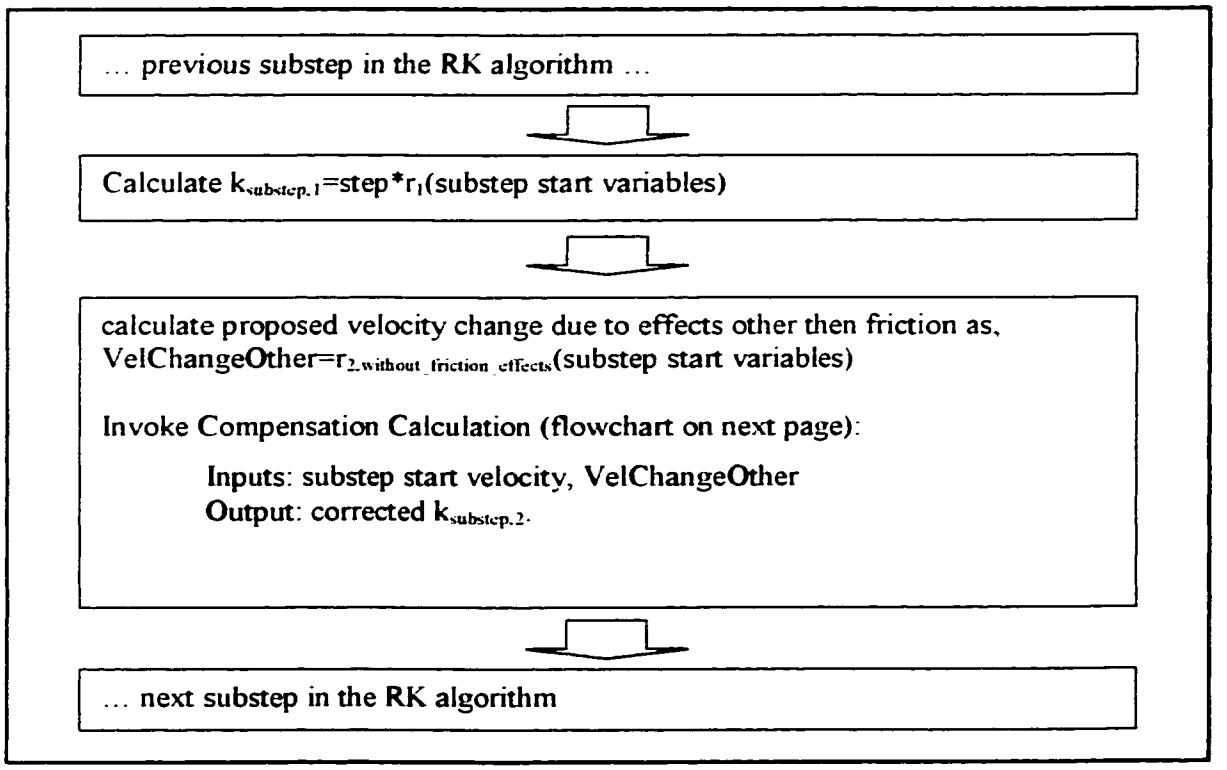


Figure 2-4: Sample Runge Kutta Routine Sub-step w/o Proposed Modification

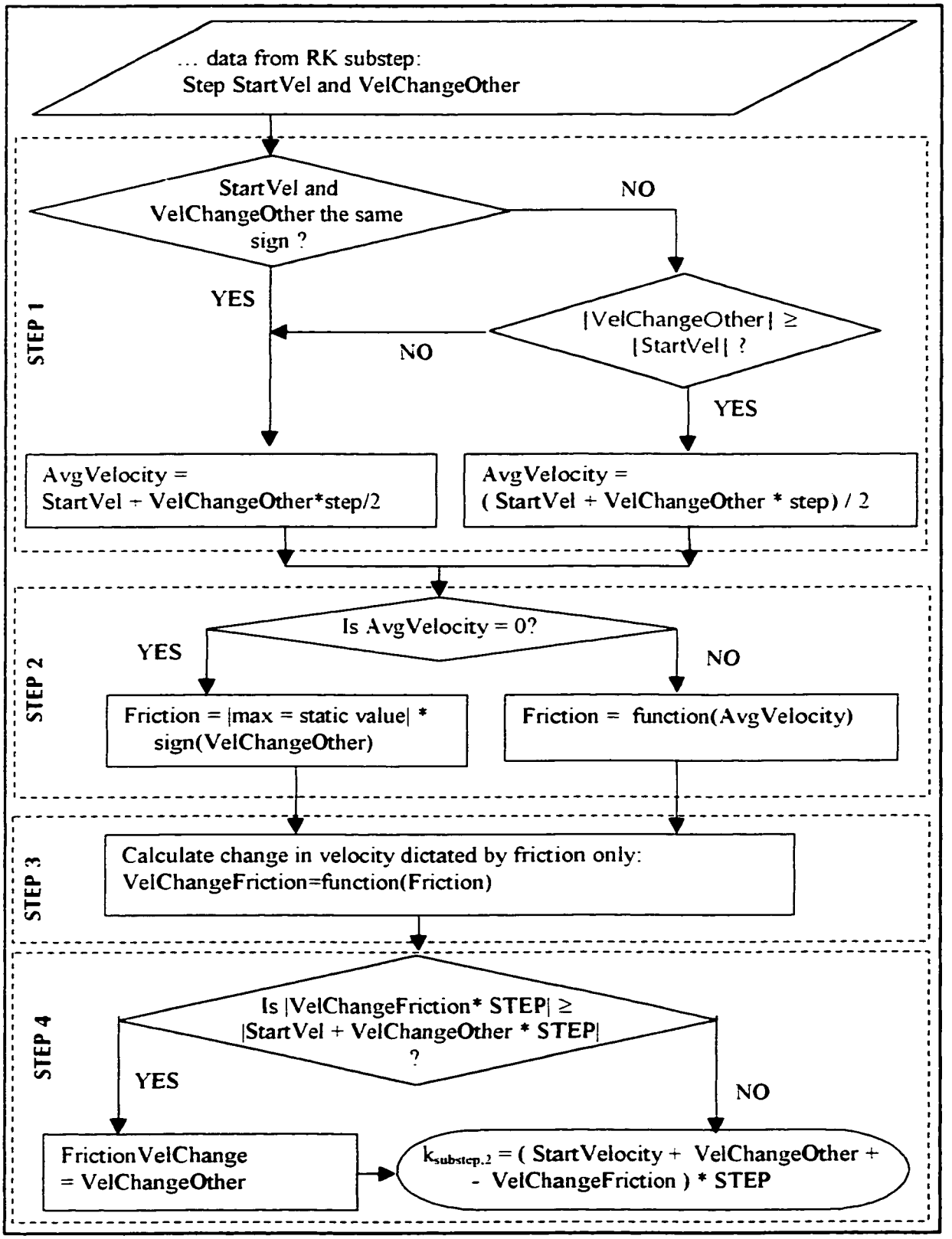


Figure 2-5: Schematic Flowchart of Compensation Routine

STEP 1: Obtain velocity at which friction is calculated

Because the friction is a function of the velocity, it is more accurate to calculate it at the sub-step <average velocity> rather than at the sub-step <start velocity> or <end velocity> - especially since the <start velocity> and <end velocity> may be of different signs, in which case, not only will the friction value be over-estimated, but also the friction may be in the incorrect direction, increasing the simulation error. With this in mind, the <average velocity> was calculated differently depending on two situations:

Recall that <proposed velocity change> refers to the velocity change that is would occur before the effects of friction was taken into account.

If the <start velocity> and the <proposed velocity change> were in the same direction OR the <start velocity> and the <proposed velocity change> were in opposite directions, but the magnitude of the <proposed velocity change> was not greater than the magnitude of the <start velocity>, i.e. if the start and the final velocity had the same sign, the <average velocity> was calculated as

<average velocity> = <start velocity> + <proposed velocity change> * <step> /2

If this was not the case, i.e. if the <proposed velocity change> would be large enough to change the sign of the velocity, the friction was calculated based on an <average velocity> calculated as

<average velocity> = (<start velocity> +<proposed velocity change> * <step>)/2,

which essentially equals half of the final velocity.

STEP 2: Calculate friction

If the above calculations dictate a zero <average velocity>, the stick-slip friction takes on the value of the static friction. The stick-slip function itself cannot be called directly to assign a value for zero velocity because at this velocity, the static friction is multi-valued, i.e. either negative or positive. The algorithm assigns the static friction value which opposes motion in the direction prescribed by the <proposed velocity change>.

If the <average velocity> is not zero, the <friction> is calculated using the stick-slip and viscous friction functions, evaluated at the <average velocity>

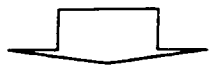


Figure 2-6: Explanatory Flowchart of Compensation Routine

STEP 3: Calculate proposed velocity change due to friction only

Similar to the calculation of the <proposed velocity change>, i.e. variable VelChangeOther in Figure (2-3), a <velocity change due friction> value is calculated using the same function, $r_2(.)$, but this time only taking into consideration the effects of the <friction> calculated in STEP 2, and nothing else. This <velocity change due friction> is assigned name VelChangeFriction in Figure (2-4).

STEP 4: Check that Proposed Velocity Change due Friction is not too much, and output

This step is introduced to ensure that the <velocity change due friction> is not the sole cause of a velocity change in sign. That is, this step checks whether the dictated <velocity change due friction> for the time period <step> is not larger in magnitude than the <start velocity + proposed velocity change>.

If this is the case, the <velocity change due friction> is set equal to the <start velocity + proposed velocity change>, insuring that the velocity is brought to zero at the end of time period <step>.

If it is not, the <velocity change due friction> is not modified.

The term $k_{substep,2}$ is then correctly calculated, including the effects of all friction functions.

NOTE that in the above, the effect of friction is evaluated for the time period of the entire <step>. Other values were tried for this time period, including the actual time period that each substep amounts to at the end (i.e. (1/6 of <step>) for first and fourth substep, and (2/6 of <step>) for second and third substeps). In such cases, the simulation algorithms depicted high frequency velocity changes with smaller amplitude than the classical Runge Kutta algorithm.

Figure 2-6 (continued): Explanatory Flowchart of Compensation Routine

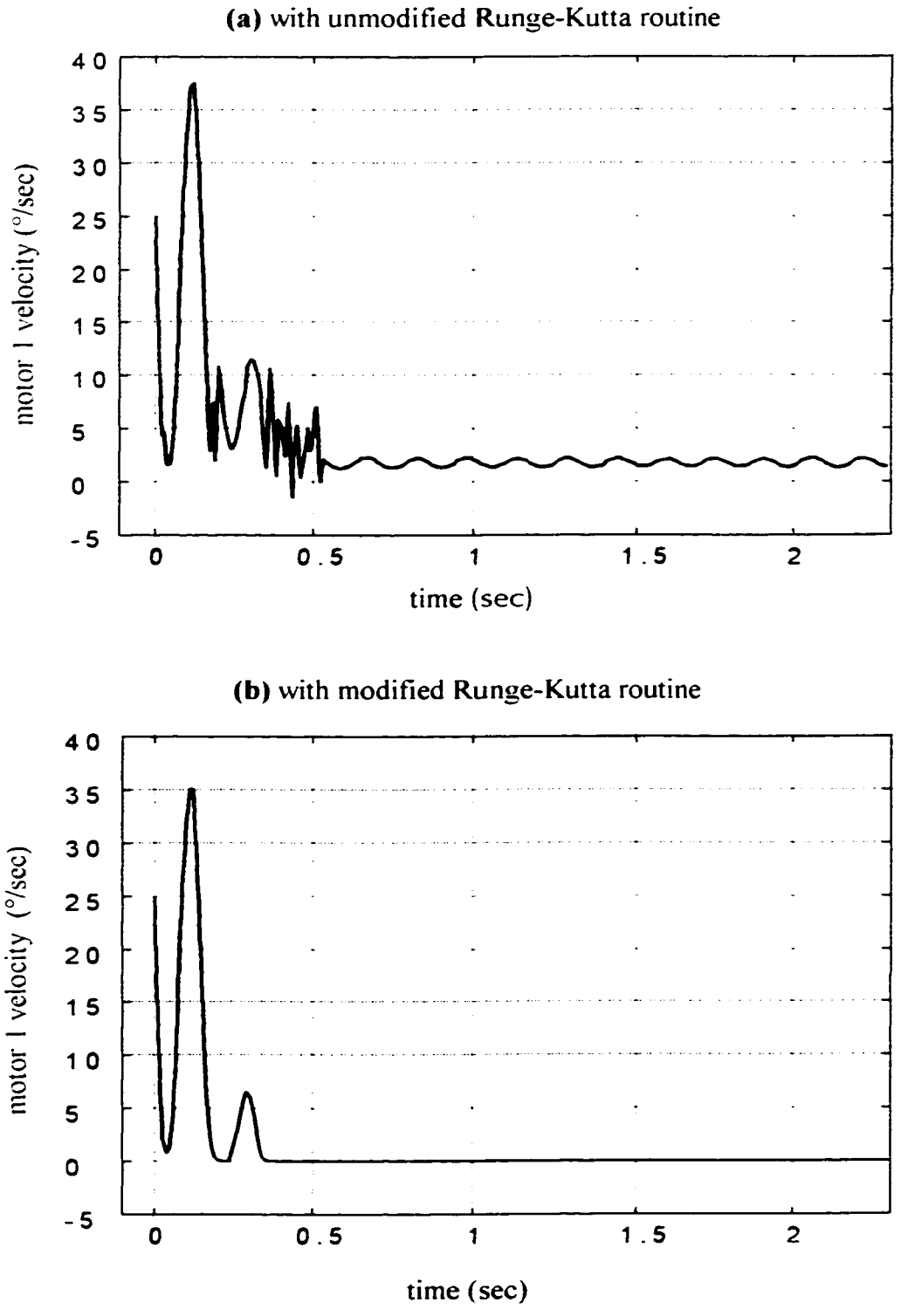


Figure 2-7: Sample Simulations Result w/o Proposed RK Modification: Velocity of Motor 1

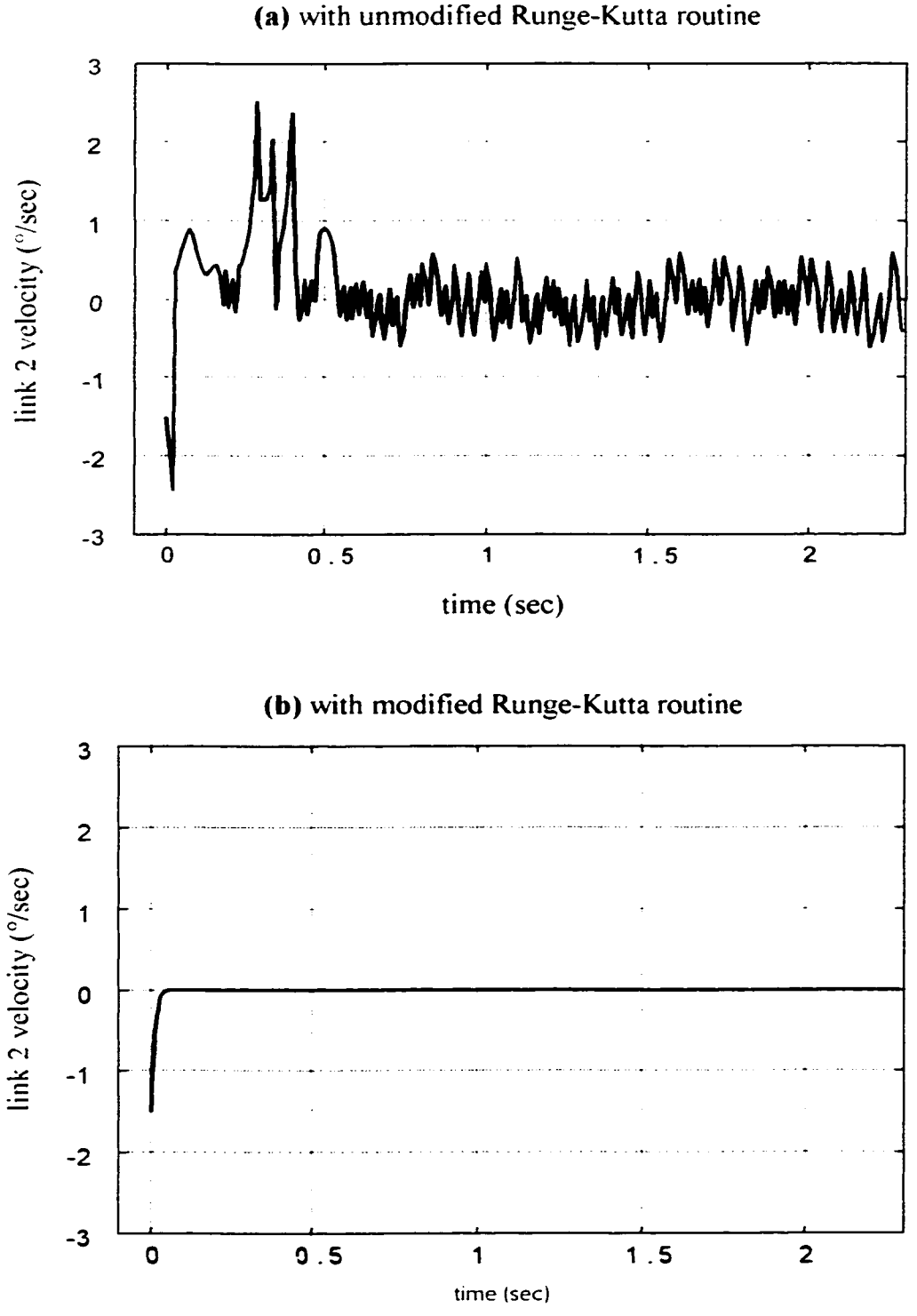


Figure 2-8: Sample Simulation Result w/o Proposed RK Modification: Velocity of Link 2

2.3.2 OPTIMIZATION PROCESS

The simulation procedure described in Section 2.3.1 is used during the optimization of the proposed linear model to compare the model under development, with the nonlinear dynamic model. For best understanding of the optimization process, simulation results for the nonlinear model and the proposed linear model at various stages during the optimization process for a sample scenario are presented in the following figures. The models included in these figures are described below, in Table (2-1).

Model	Description
model #1	the best linear model available in the literature, the linearized version of model (2-2) -(2-3)
model #2	a version of the proposed linear model (2-16)-(2-17) with all the stick-slip friction derivatives set to 0. The only difference between this model and model #1 is that this model takes into account the Λ term of Equations (2-5)-(2-6), i.e. the effect of the velocities of links *1..i*, and rotor *1..i*, on the kinetic energy of rotor *j*, $i \neq j$.
model #3	a version of the proposed linear model (2-16)-(2-17) with the stick slip friction derivatives set to : $\frac{\partial \mathbf{F_{CM}}}{\partial \mathbf{x_2}} = \begin{bmatrix} 3.5 & 0 \\ 0 & 2.5 \end{bmatrix}$, $\frac{\partial \mathbf{F_{CL}}}{\partial \mathbf{x_4}} = \begin{bmatrix} 1.75 & 0 \\ 0 & 0.75 \end{bmatrix}$
model #4	a version of the proposed linear model (2-16)-(2-17) with the stick slip friction derivatives set to : $\frac{\partial \mathbf{F_{CM}}}{\partial \mathbf{x_2}} = \begin{bmatrix} 14 & 0 \\ 0 & 10 \end{bmatrix}$, $\frac{\partial \mathbf{F_{CL}}}{\partial \mathbf{x_4}} = \begin{bmatrix} 7 & 0 \\ 0 & 3 \end{bmatrix}$
model #5	the nonlinear dynamic model, taken as the standard for comparison., given by Equations (2-5)-(2-15)

Table 2-1: Sample Models Simulated During the Optimization Process

The linearization point for the linear models was taken as [90 10 0 0 90 10 0 0].

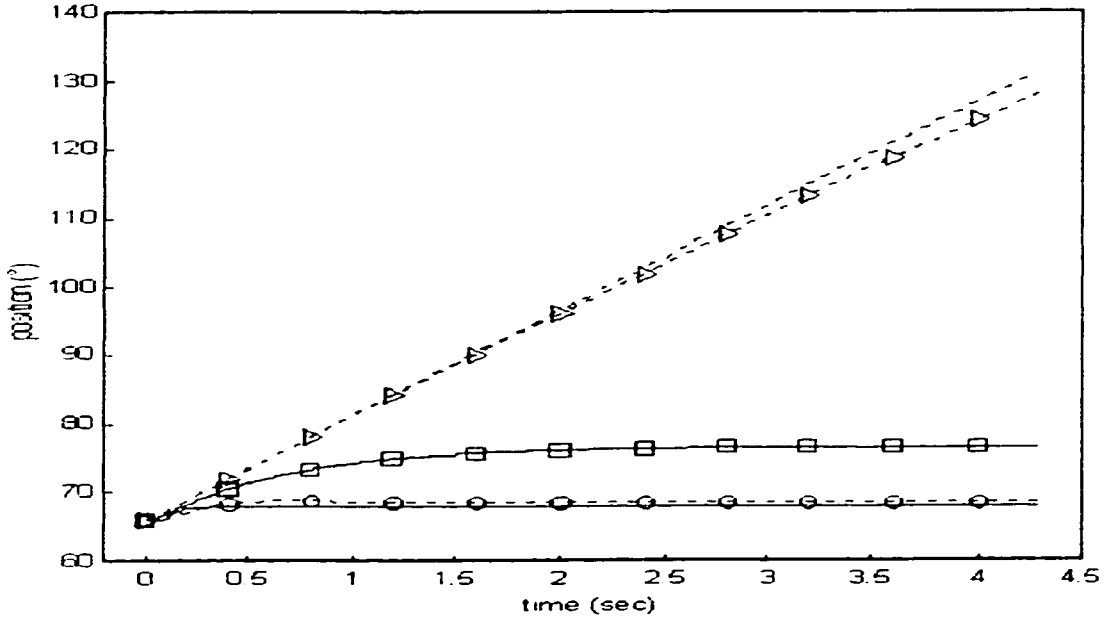
Discussion of Results:

The best linear model available in the literature, model #1, is shown to not provide adequate tracking of the nonlinear dynamic model, model #2, for any of the eight states observed. The steady-state error is too high, and/or the settling time is too long. Model #2 is introduced to show that including the effects of the velocities of links *1..i* and rotor *1..i* on the kinetic energy of rotor *j*, *i* *j*, results in negligible improvements in the linear dynamic model. Model #2 is still unacceptable. Model #3 also provides unacceptable tracking of the nonlinear dynamic model. However, it can be observed that the introduction of positive values for the derivatives of the stick-slip frictions has dramatically increased the tracking of the nonlinear dynamic model by the linear dynamic model. Model #4 is taken as the final proposed linear dynamic model. This linear model provides the best tracking of the nonlinear dynamic model.

For higher values of the derivatives of the stick-slip friction, the system behavior deteriorates. The difficulty of this optimization lies in finding the optimum values that provide best results for all eight states of the system, for all operating scenarios. While this section presents the results for only one such scenario, during the actual optimization process, the results of several such scenarios were monitored.

Model #4 of this section is taken as the final proposed linear model. However, in order for this model to be allowed to be used in controller design instead of the nonlinear model, it must now be shown that this model provides acceptable tracking of the nonlinear dynamic model for all operating conditions. This is achieved next.

(a) state 1: motor 1 position



(b) state 2: motor 2 position

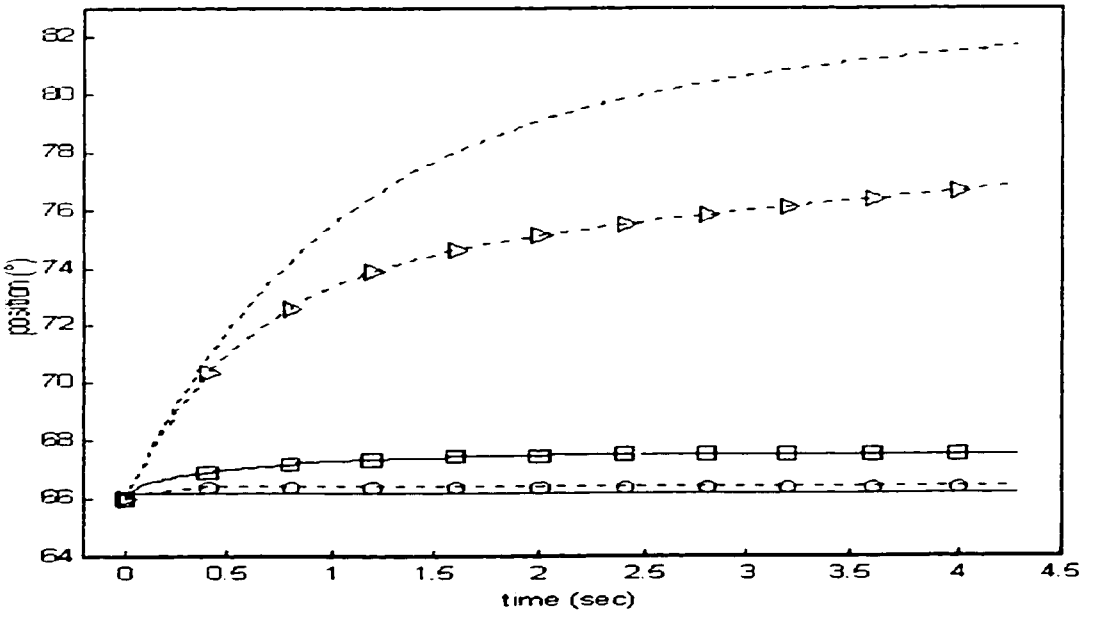
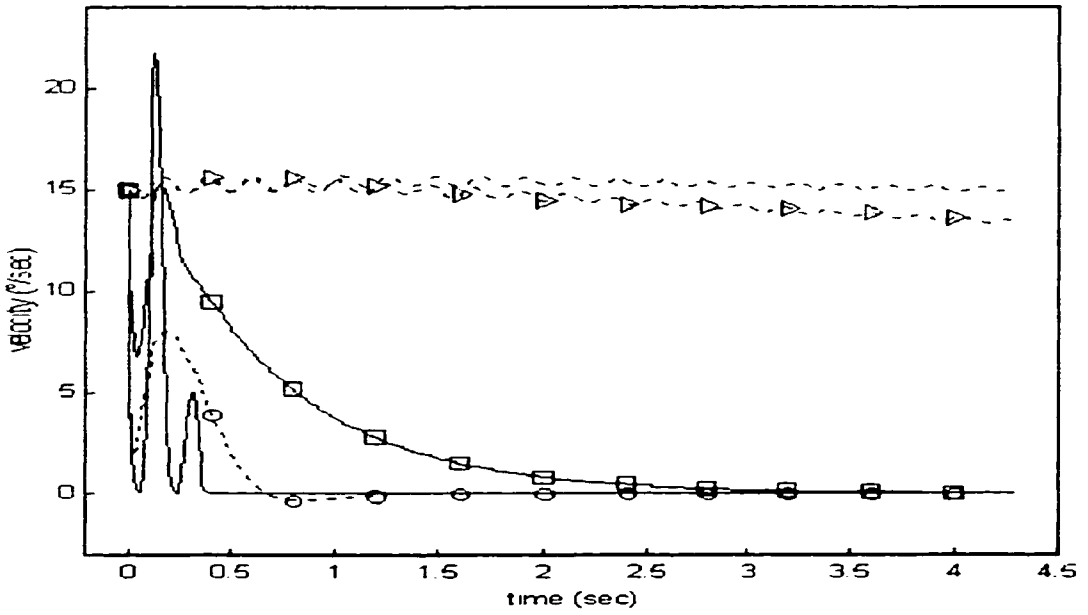


Figure 2-9: Optimization of the Proposed Linear Model

model 1 ---------- model 2 -Δ-----Δ- model 3 -□——□-
model 4 -⊖-----⊖- model 5 ————

(c) state 3: motor 1 velocity



(d) state 4: motor 2 velocity

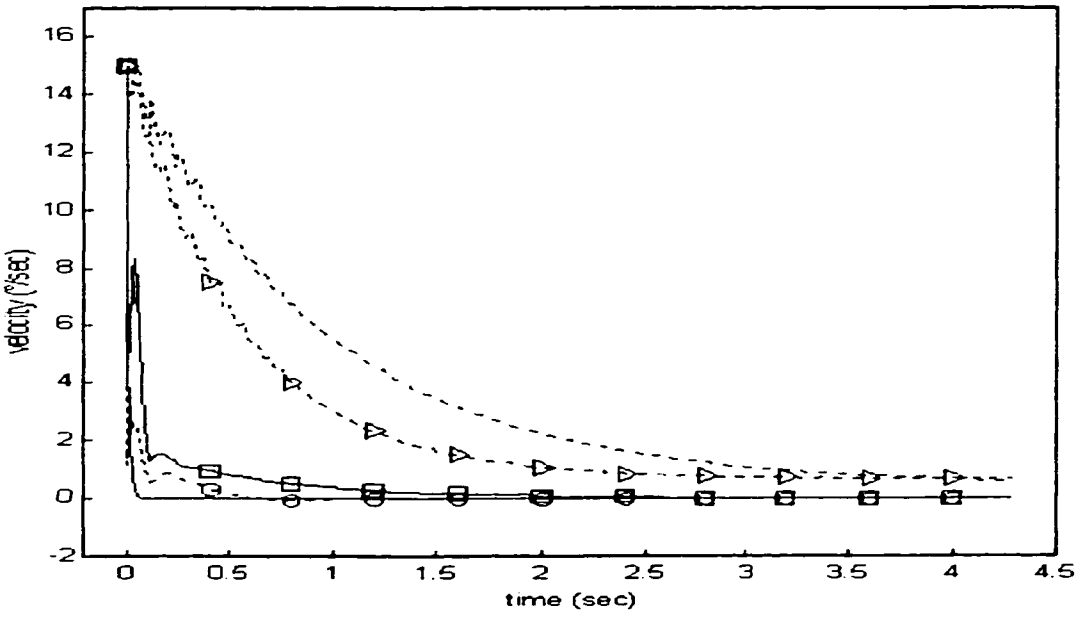
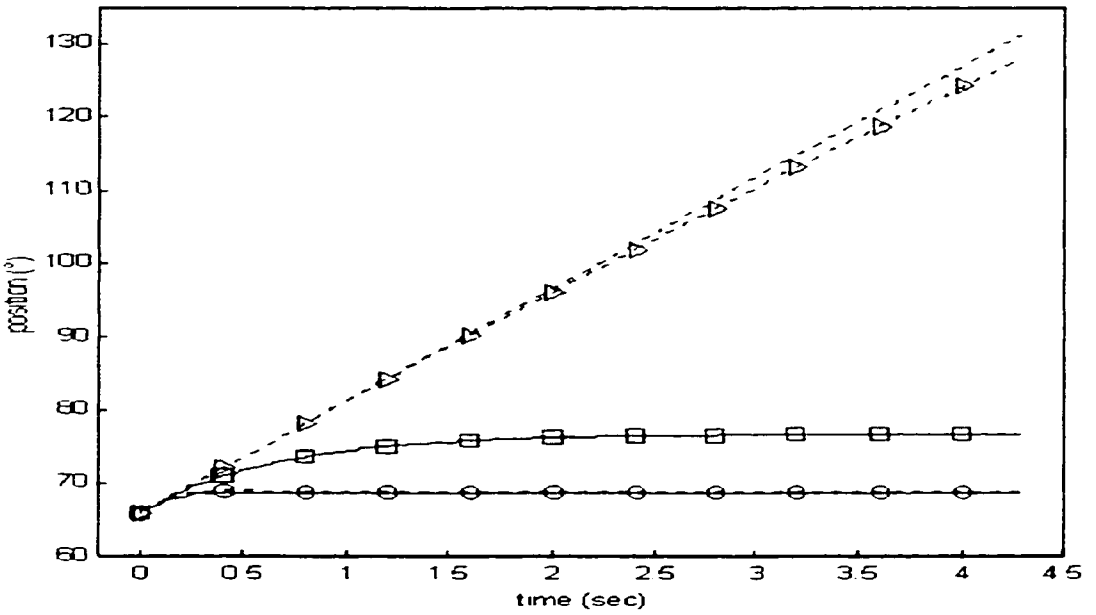


Figure 2-9: Optimization of Proposed Linear Model (continued)

model 1 ---------- model 2 -Δ---Δ- model 3 -□---□-
model 4 -⊖----⊖- model 5 ————

(e) state 5: link 1 position



(f) state 1: link 2 position

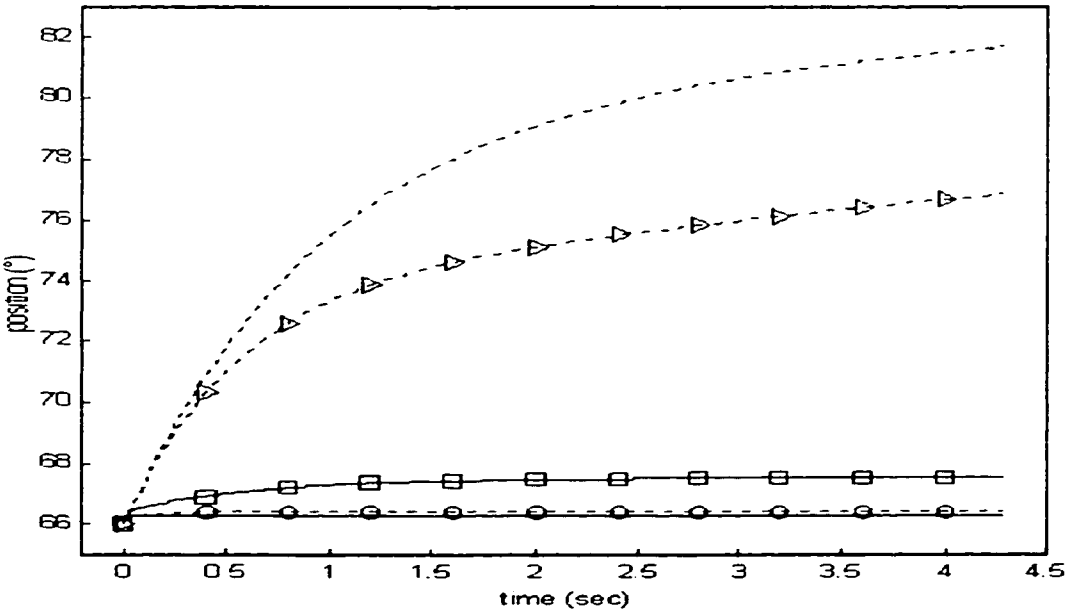
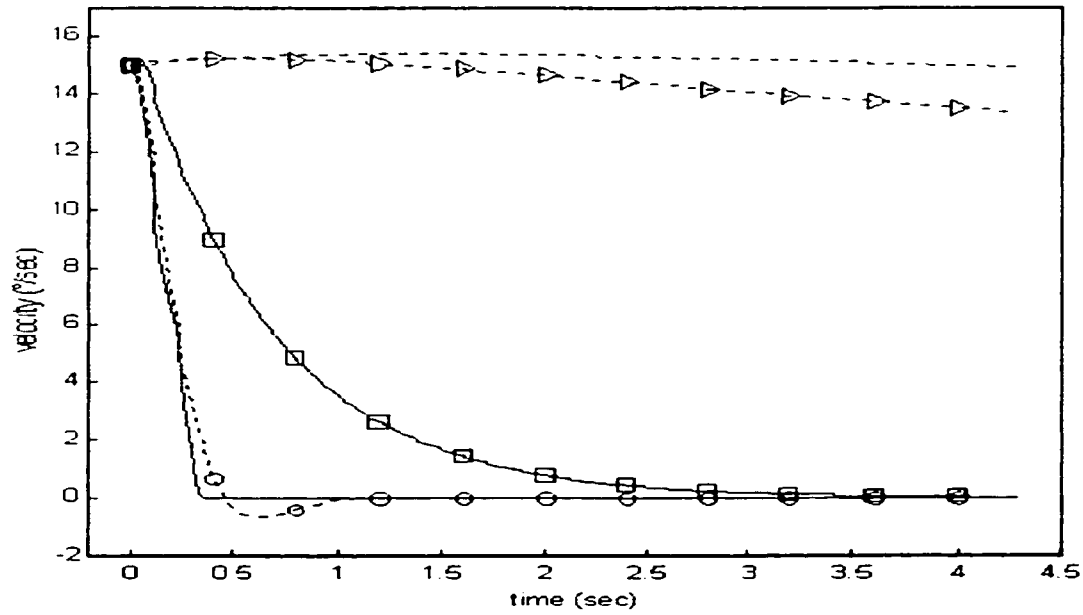


Figure 2-9: Optimization of Proposed Linear Model (continued)

model 1 ---------- model 2 —Δ—Δ— model 3 —□—□—
model 4 —○—○— model 5 ————

(f) state 7: link 1 velocity



(g) state 8: link 2 velocity

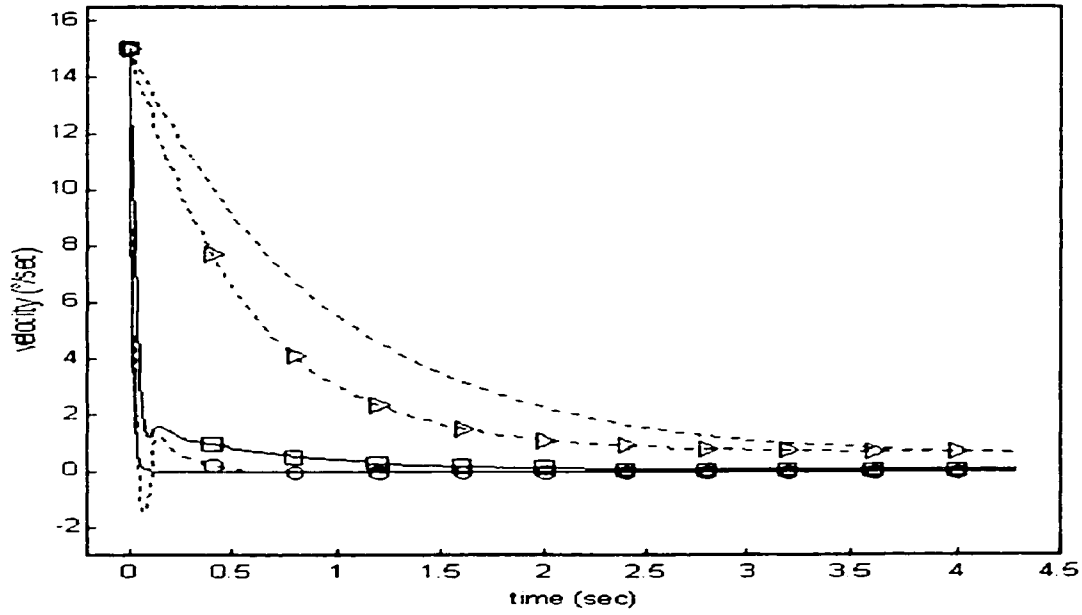


Figure 2-9: Optimization of Proposed Linear Model (continued)

model 1 ---------- model 2 —Δ—Δ— model 3 —□—□—
model 4 --○-----○-- model 5 ————

2.3.3 MODEL VALIDATION

This section contains simulations results which compare the time response of the proposed linear dynamic model with the time response of the nonlinear dynamic model. It would take a vary long amount of time to perform this comparison for every possible scenario (set of initial conditions). In this dissertation the results of three such scenarios are presented. These scenarios are described in Table (2-2). Remembering that the linearization point (aside from the values of the stick-slip function derivatives) was chosen as [90 10 0 0 90 10 0 0], scenario #1 describes a set of initial conditions fairly distant from the linearization point, while scenario #2 describes a set of initial conditions fairly close to the linearization point. Scenario #3, which in fact contains initial conditions values outside the actual operating range of the robot, was chosen to check the validity of the proposed linear dynamic model in a worst case scenario, trigonometrically very different from the linearization point. Since it is more difficult for the linear model to accurately track the nonlinear dynamic model at high speeds, only high link and motor speeds were used as initial conditions in these scenarios.

state	scenario #1	scenario #2	scenario #3
$x_1(°)$	10.0	97.0	-170.0
$x_2(°)$	-81.0	-7.1	160.0
$x_3(°/s)$	25.0	25.0	25.0
$x_4(°/s)$	25.0	-25.0	-15.0
$x_5(°)$	10.2	97.0	-170.0
$x_6(°)$	-80.9	-7.10	160.2
$x_7(°/s)$	25.0	25.0	25.1
$x_8(°/s)$	25.2	-25.1	-15.1

Table 2-2: Initial Conditions Used in the Three Sample Scenarios

The simulation results for the three scenarios are given in the plots of Figures (2-10), (2-11), and (2-12).

Discussion Of Results

These plots indicate that there is zero difference between the steady-state velocity values of the linear model and its corresponding nonlinear model. This is expected for this type of simulation, where the system is driven only by its initial conditions. Also expected and acceptable is the error between the linear and the corresponding nonlinear dynamic model during the transient stage. These are not measures of the tracking performance of the linear model versus the corresponding nonlinear model. The measure for evaluating tracking performance is how fast and how well the linear model tracks the nonlinear model. The multitude of research performed in this area indicates that, to be considered an acceptable representation of the nonlinear model, the linear model must achieve steady-state no later than 2 seconds after the nonlinear model has achieved steady-state, with an error of no more than 5°. The steady-state error is not expected to be 0 because it is understood that the linear model is not an exact match for the nonlinear dynamic model.

The results of scenario #1, shown in Figure (2-10), indicate excellent tracking. All plots show that the linear model reaches steady state less than 1 second after the nonlinear model reaches steady state, with a maximum steady-state error of 1.3°.

The results of scenario #2, shown in Figure (2-11), again indicate excellent tracking by the proposed linear model of the corresponding nonlinear dynamic model, both in terms of steady-state position differences and time to reach steady state. The *link*

1 position time response shows the largest steady-state error, of all three scenarios, of approximately 2.0°. All plots indicate that the linear model tracks the nonlinear model with a maximum time difference of 1 second.

In scenario #3, Figure (2-12), although the position variables have initial values at the extreme of their operating range, the results shows even better tracking performance. The maximum steady-state error is less than 0.8°, and the linear model tracks the nonlinear model with a time difference of 1 second.

Beside the three scenarios described in this dissertation, more than ten other scenarios were checked during the actual work. Their results were similar to those of the three scenarios presented.

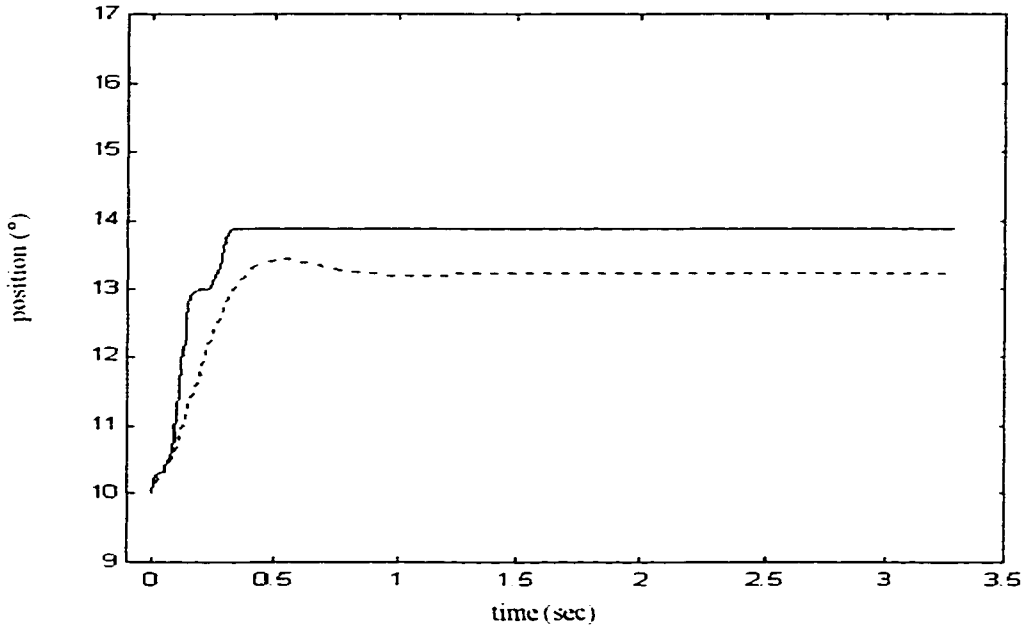
Summary Of Results

In general, linear models provide accurate tracking of the corresponding nonlinear model only for operation near the linearization point or at low speeds. However, the scenarios presented in this section prove that the proposed linear model provides accurate tracking of the nonlinear model over the entire allowed position space, at fairly high speeds. This constitutes the validation of the proposed linear model.

2.3.4 THE PLANT DISTURBANCE FACTOR

Applying Equation (2-22) to each state in each simulation scenario results in the values of c outlined in Table (2-3). Note that, due to its intended use in Equation (2-18), the units of c are meaningless. However, it is necessary to indicate that these units are radian-based (e.g. radians or radians/second), and not degree-based.

(a) state1: motor 1 position



(b) state2: motor 2 position

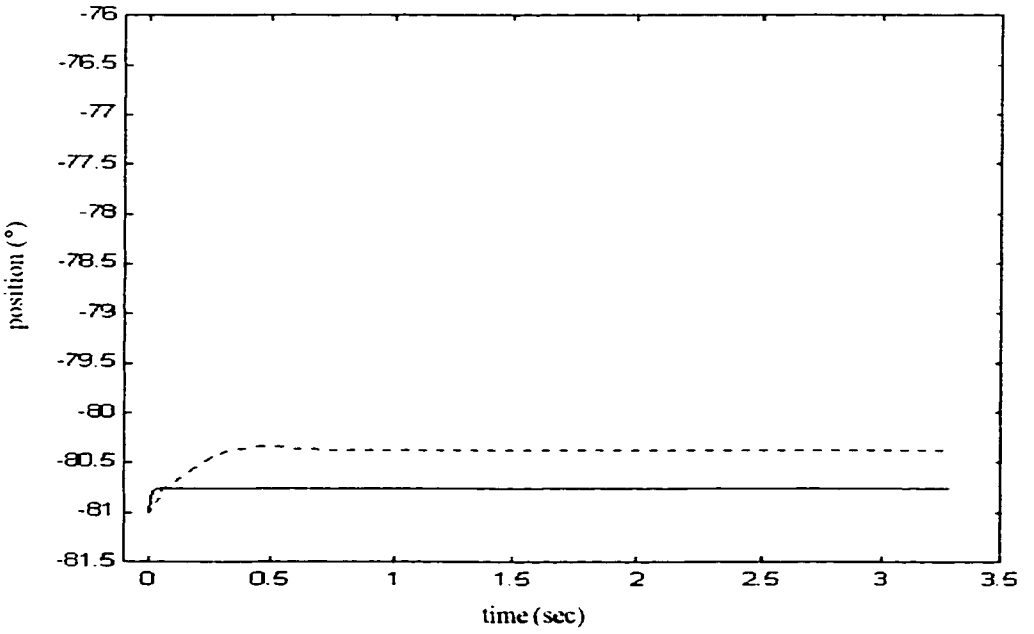
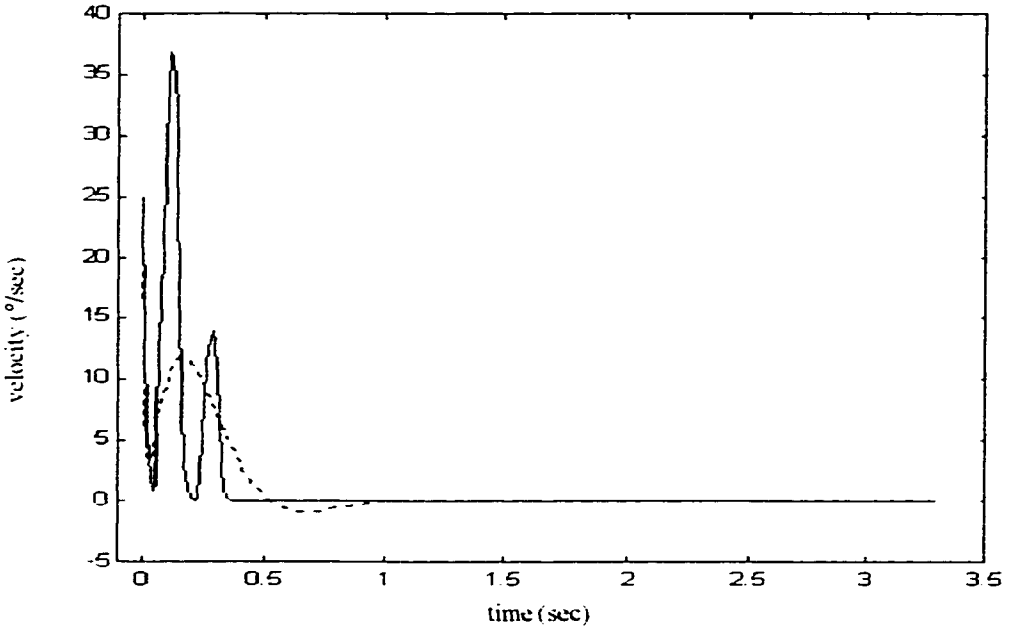


Figure 2-10: Model Validation, Scenario #1
nonlinear (solid), linear proposed (dashed)

(c) state3: motor 1 velocity



(d) state4: motor 2 velocity

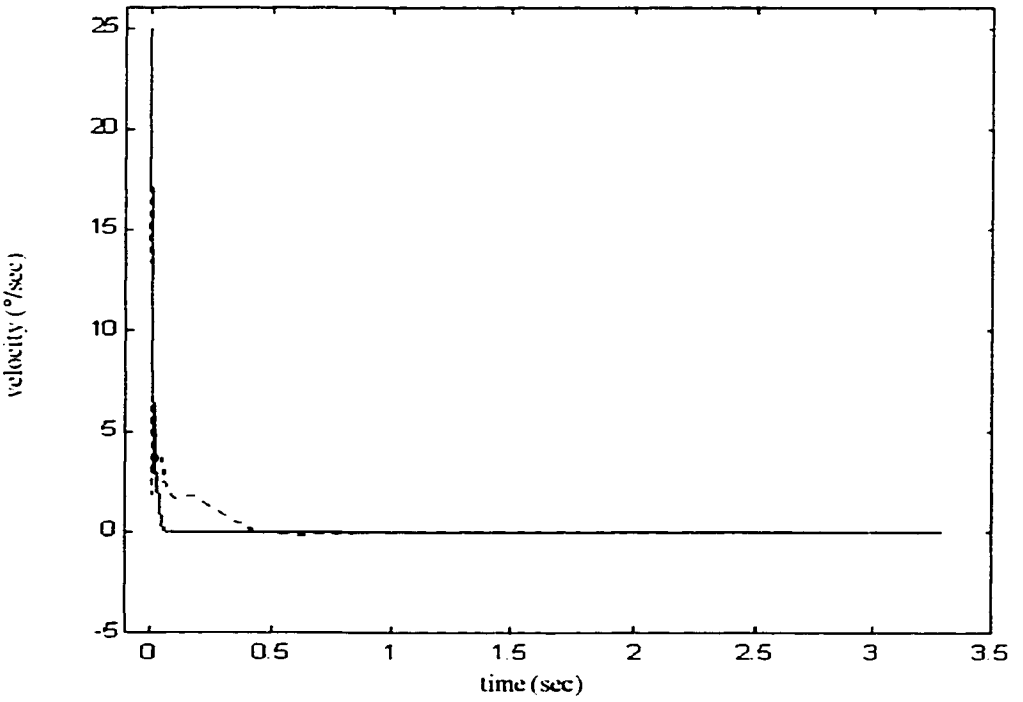
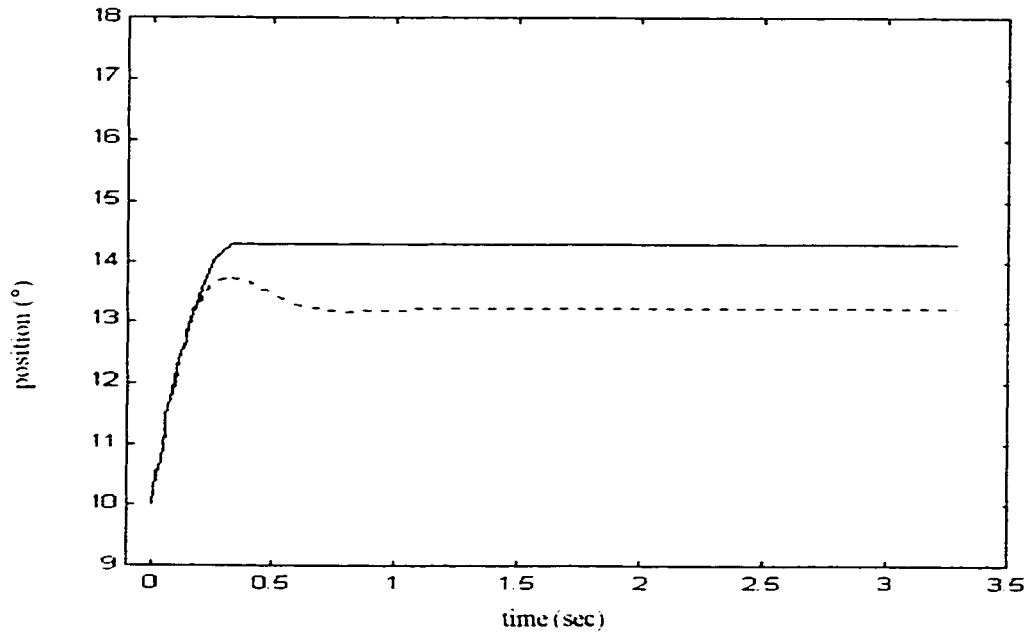


Figure 2-10: Model Validation, Scenario #1 (continued)
nonlinear (solid), linear proposed (dashed)

(e) state5: link 1 position



(f) state 6: link 2 position

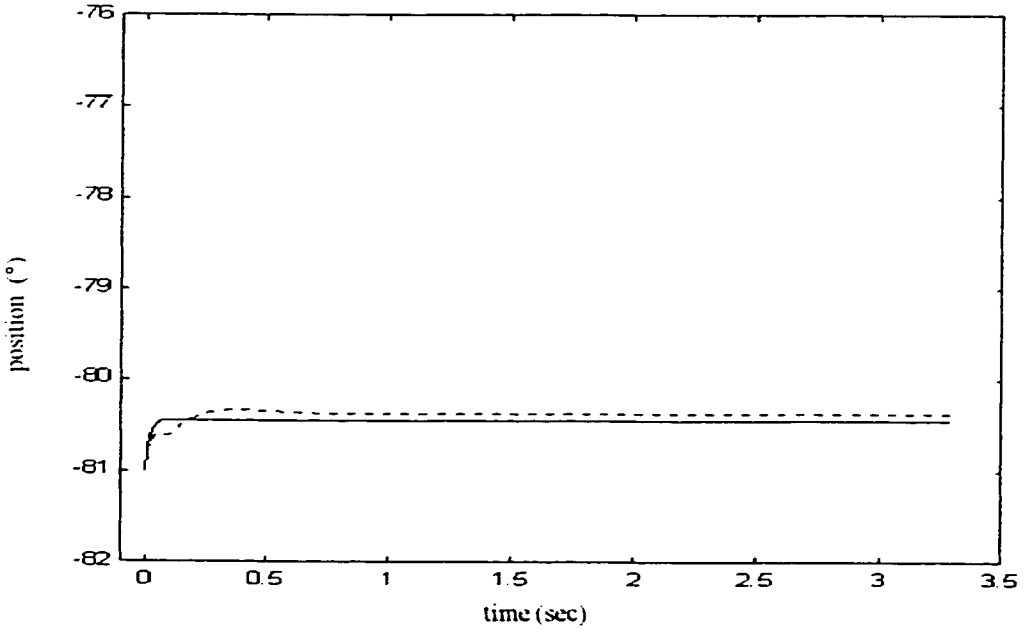
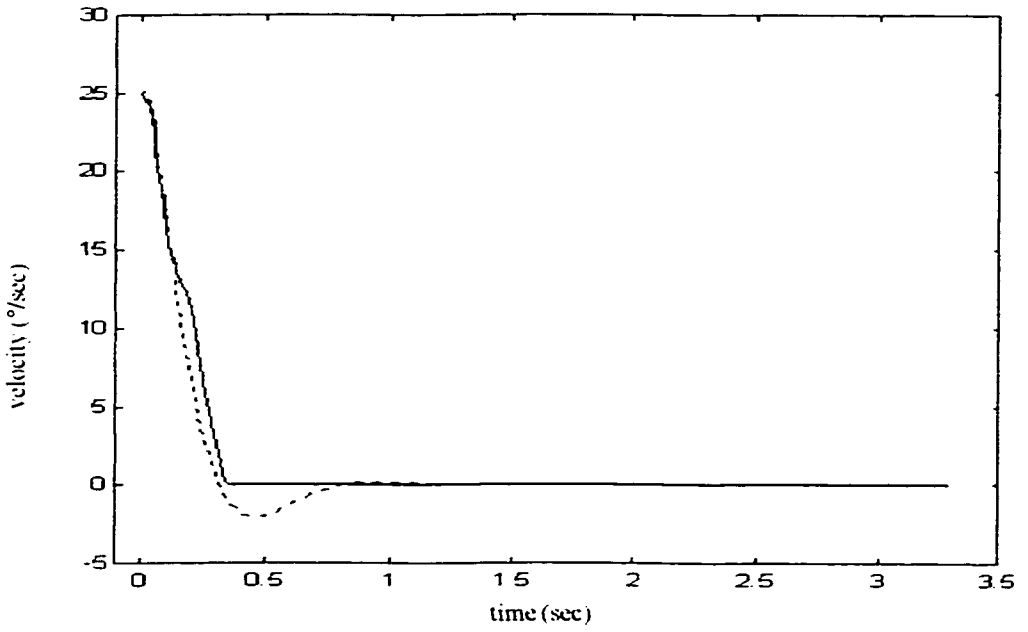


Figure 2-10: Model Validation, Scenario #1 (continued)
nonlinear (solid), linear proposed (dashed)

(g) state7: link 1 velocity



(h) state8: link 2 velocity

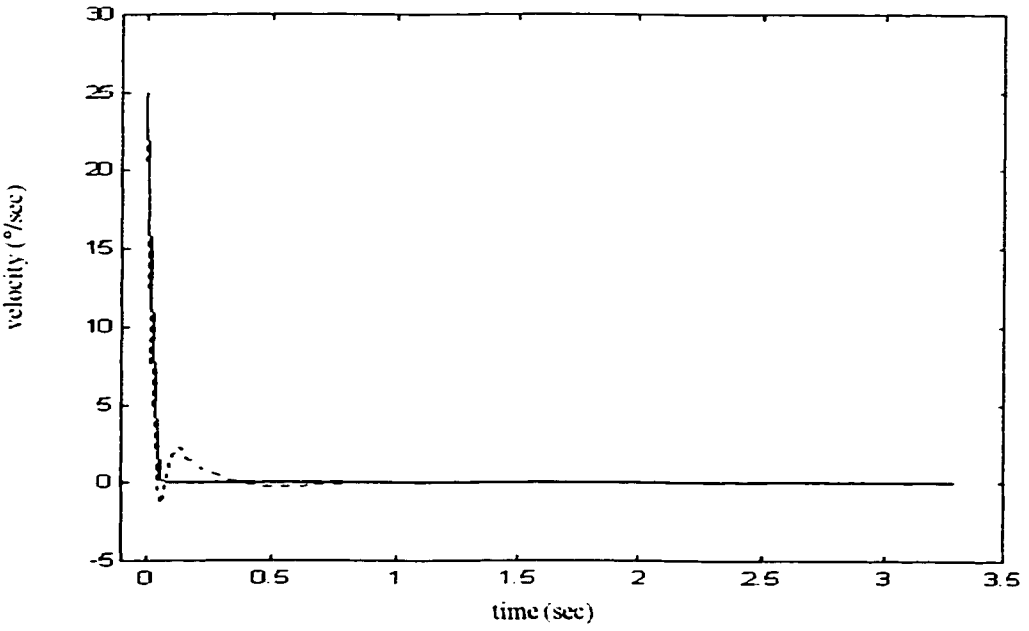
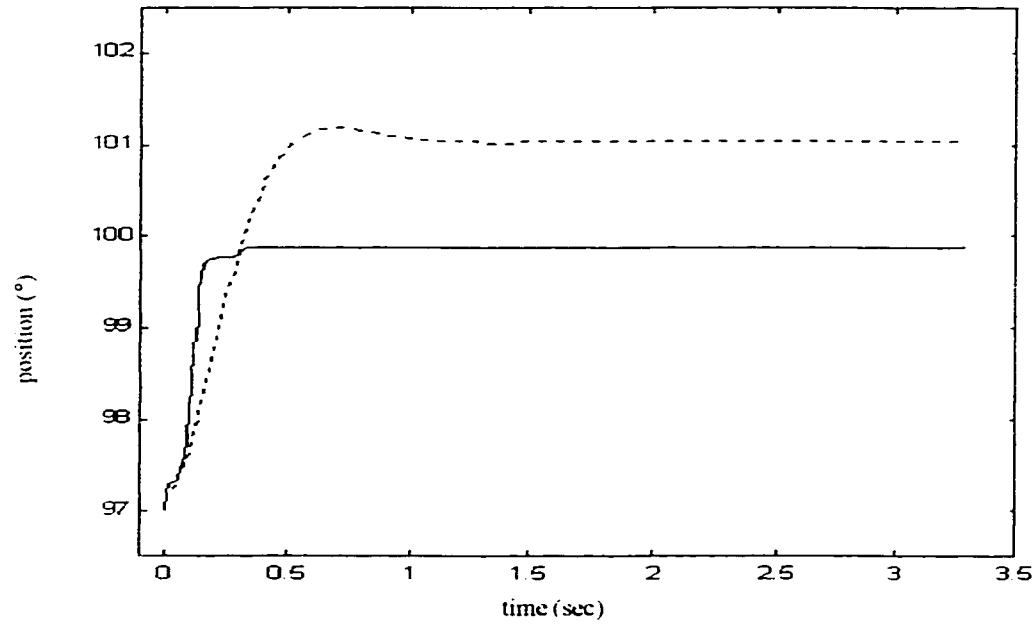


Figure 2-10: Model Validation, Scenario #1 (continued)
nonlinear (solid), linear proposed (dashed)

(a) state1: motor 1 position



(b) state2: motor 2 position

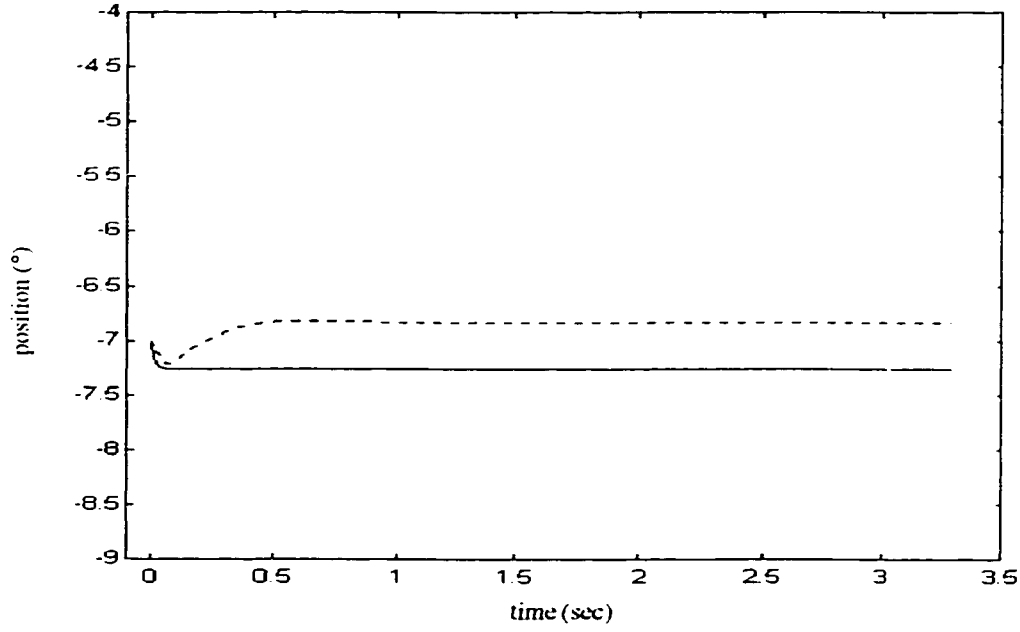
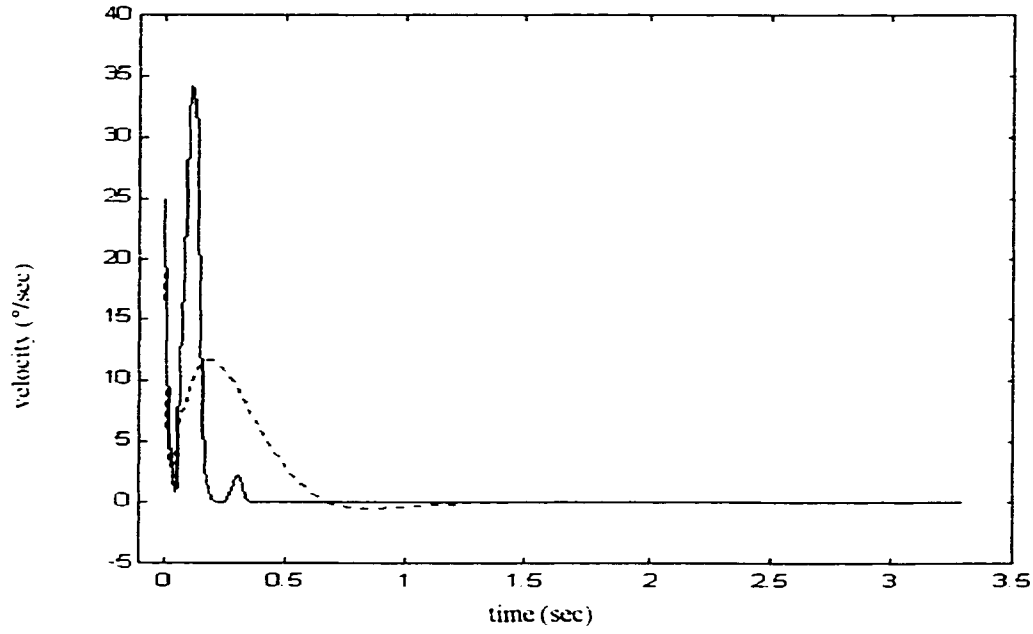


Figure 2-11: Model Validation, Scenario #2
nonlinear (solid), linear proposed (dashed)

(c) state3: motor 1 velocity



(d) state4: motor 2 velocity

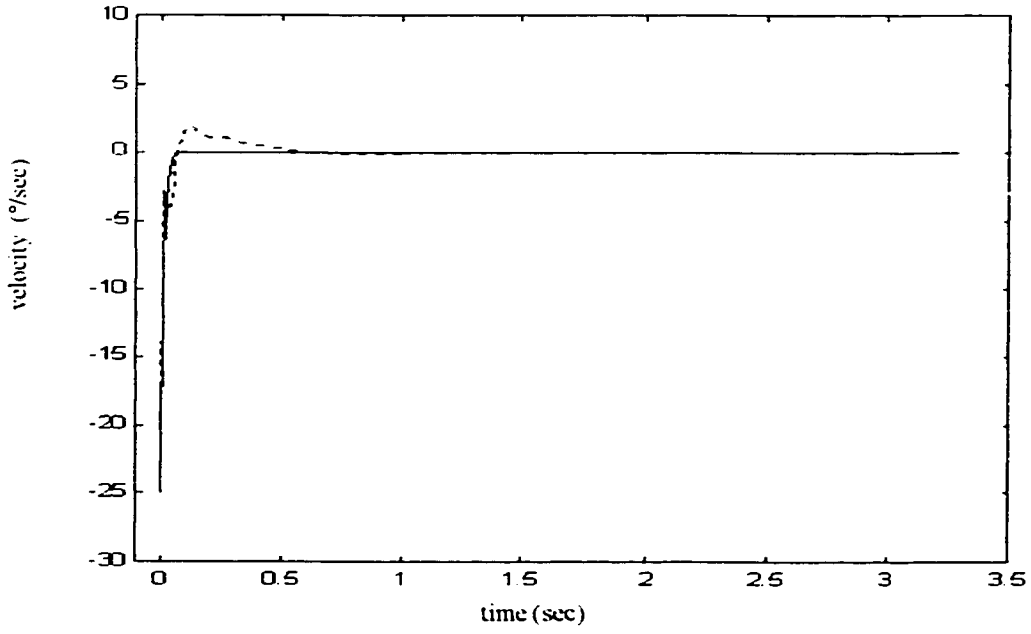


Figure 2-11: Model Validation, Scenario #2 (continued)
nonlinear (solid), linear proposed (dashed)

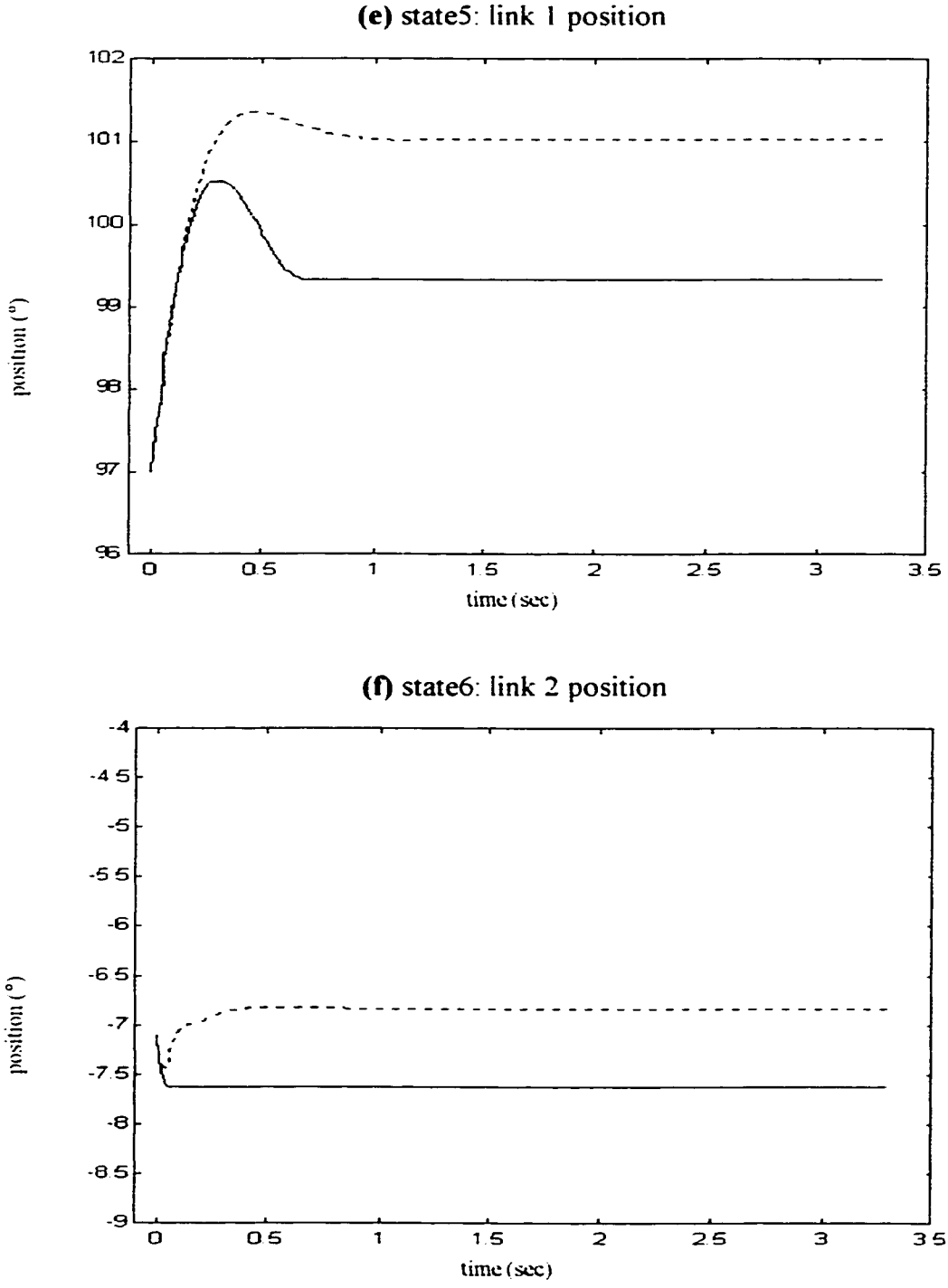
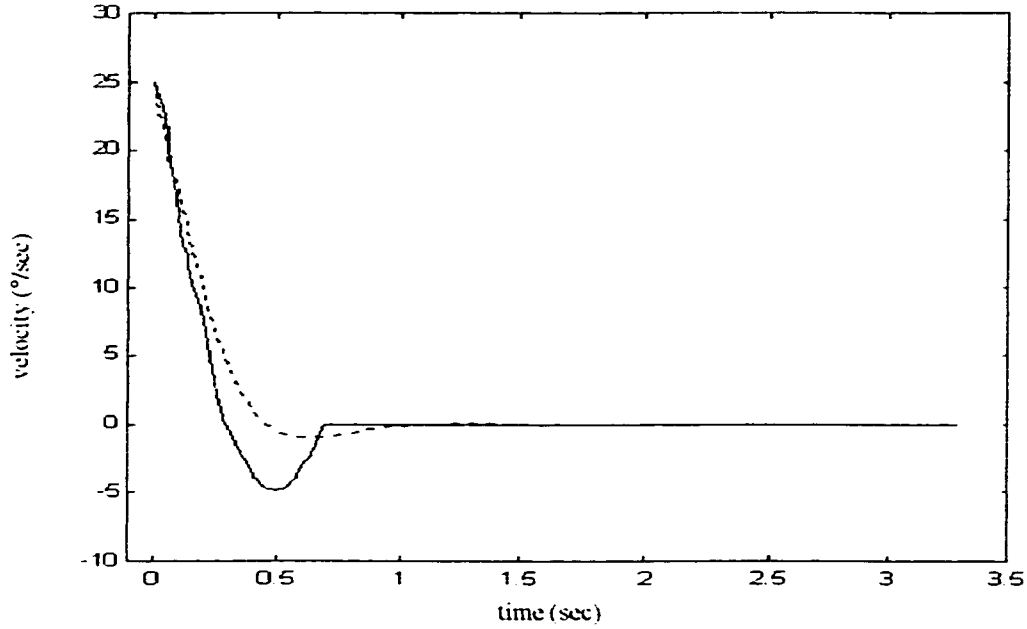


Figure 2-11: Model Validation, Scenario #2 (continued)
nonlinear (solid), linear proposed (dashed)

(g) state7: link 1 velocity



(h) state8: link 2 velocity

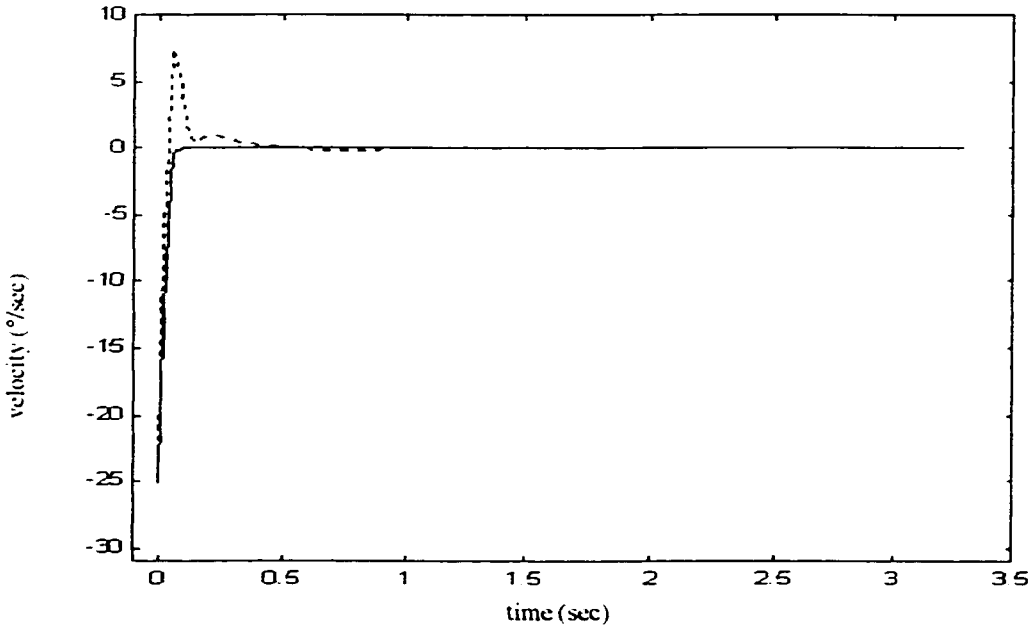


Figure 2-11: Model Validation, Scenario #2 (continued)
nonlinear (solid), linear proposed (dashed)

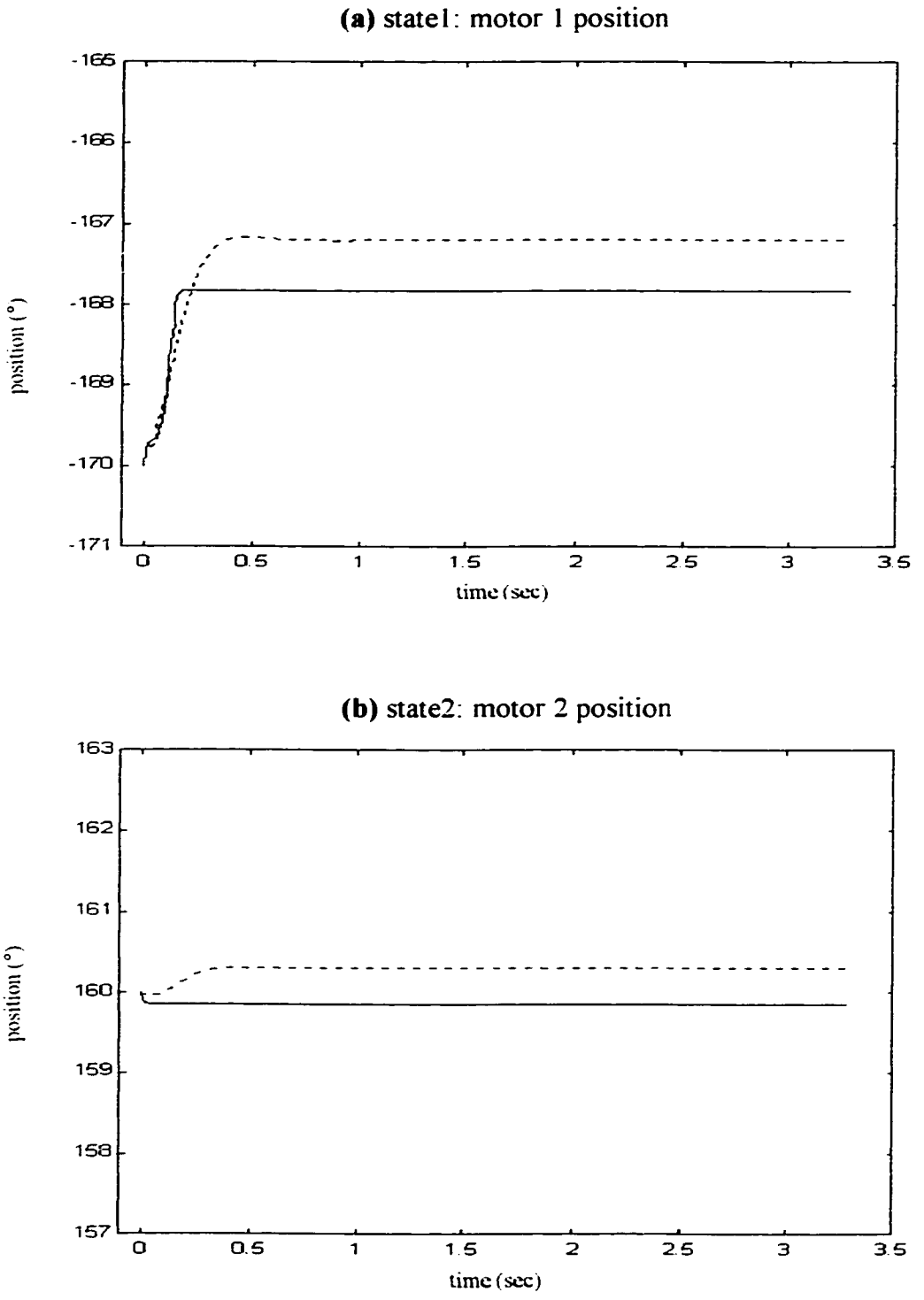
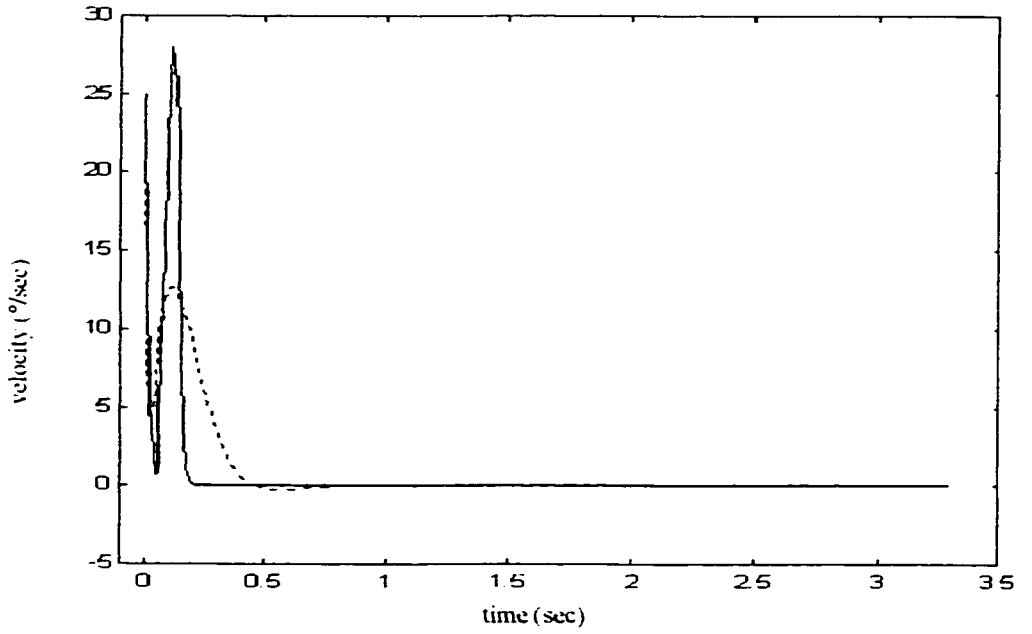


Figure 2-12: Model Validation, Scenario #3
nonlinear (solid), linear proposed (dashed)

(c) state3: motor 1 velocity



(d) state4: motor 2 velocity

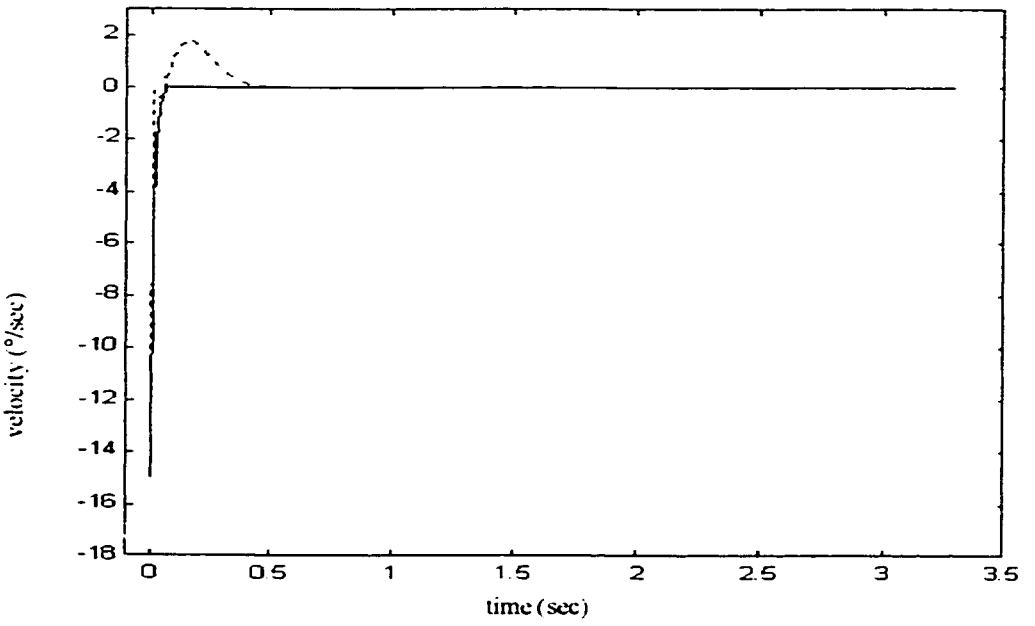
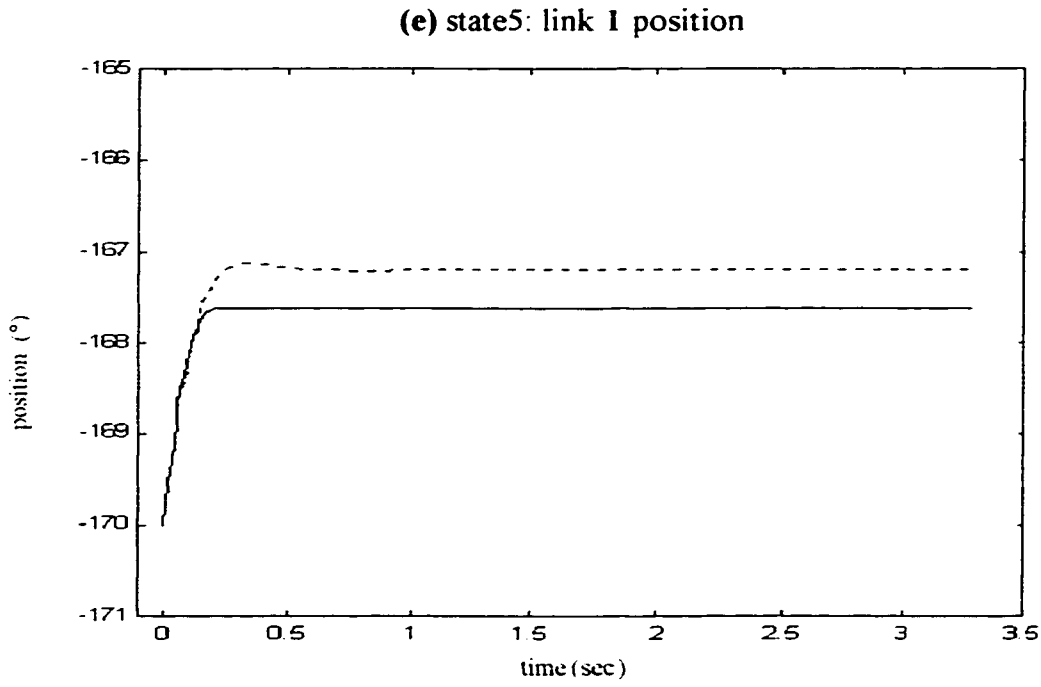


Figure 2-12: Model Validation, Scenario #3 (continued)
nonlinear (solid), linear proposed (dashed)



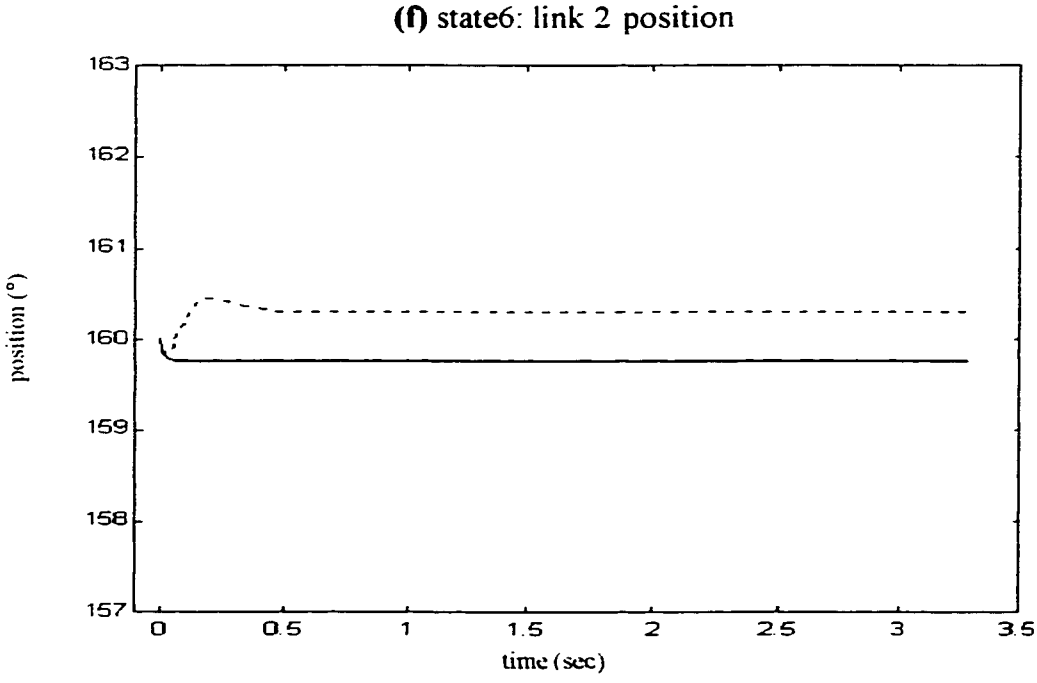
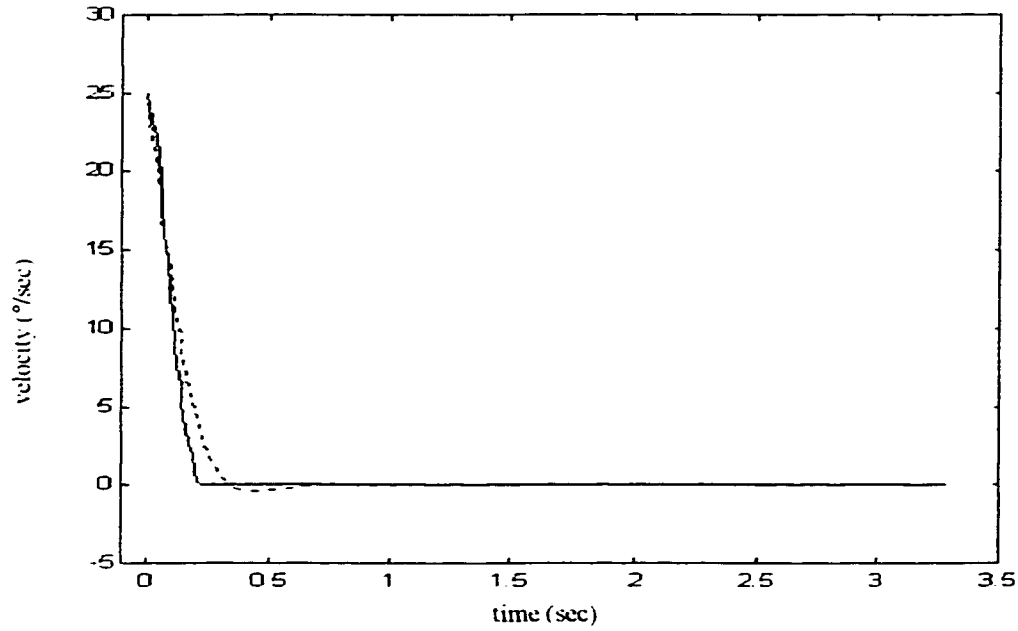


Figure 2-12: Model Validation, Scenario #3 (continued)
nonlinear (solid), linear proposed (dashed)

(g) state7: link 1 velocity



(h) state8: link 2 velocity

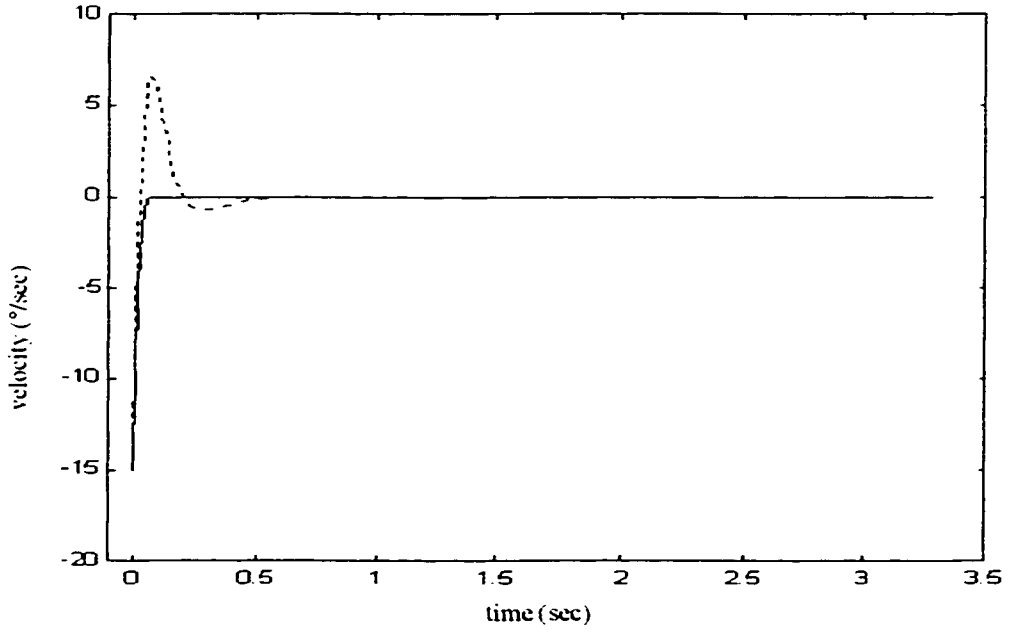


Figure 2-12: Model Validation, Scenario #3 (continued)
nonlinear (solid), linear proposed (dashed)

2.4 PROPOSED LINEAR MODEL FOR THE STUDY-CASE ROBOT

In summary, the proposed linear dynamic model for the study-case flexible-joint robot is:

$$\dot{\mathbf{X}} = \mathbf{A}\,\mathbf{X} + \mathbf{B}\,\mathbf{U} + \mathbf{G}\,\mathbf{W} \tag{2-26}$$

where,

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1621.7 & 0 & -115.2 & 0 & 1621.7 & 0 & 0 & 0 \\ -105.91 & -3005.1 & 0 & -606.58 & 105.91 & 300.1 & 3.76 & -3.87 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 105.91 & -36.535 & 0 & 5.389 & -105.91 & 36.53 & -3.76 & 3.87 \\ -247.8 & 664.85 & 0 & -12.61 & 247.83 & -664.8 & 8.80 & -44.2 \end{bmatrix}, \tag{2-27}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 8.17 & 0 \\ 0 & 60.06 \\ 0 & 0 \\ 0 & 0 \\ 0 & -0.533 \\ 0 & 1.25 \end{bmatrix} \tag{2-28}$$

and **GW** is given by

$$\mathbf{GW} = 0.076 * \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} N(0,1) \\ N(0,1) \\ N(0,1) \\ N(0,1) \\ N(0,1) \\ N(0,1) \\ N(0,1) \\ N(0,1) \end{bmatrix} \tag{2-29}$$

$$\mathbf{GW} = \begin{bmatrix} 0 \\ 0 \\ 0532\ \text{rad/s/s} \\ 0.380\ \text{rad/s/s} \\ 0 \\ 0 \\ 0.456\ \text{rad/s/s} \\ 0.456\ \text{rad/s/s} \end{bmatrix} * \mathrm{N}(0,1) = \begin{bmatrix} 0 \\ 0 \\ 30.5\ °/\text{s/s} \\ 21.8\ °/\text{s/s} \\ 0 \\ 0 \\ 26.1\ °/\text{s/s} \\ 26.1\ °/\text{s/s} \end{bmatrix} * \mathrm{N}(0,1) \qquad (2\text{-}30)$$

where N(0,1) is a Gaussian noise with zero mean and unit variance.

2.5 CONCLUSIONS

This chapter presented a new approach for modeling flexible-joint robot manipulator systems. This approach takes into consideration the effects of the velocities of the precedent links and rotors of a particular rotor, on that rotor's kinetic energy, the effects of the nonlinearity of the joint flexibility, the effects of stick-slip friction. A nonlinear dynamic model was first proposed, followed by a linear dynamic model. This linear dynamic model was derived through linearization of the proposed nonlinear dynamic model. By doing so, all the limits of the classical linear model were overcome. A stochastic term that accounts for plant disturbances was also derived and included in the linear dynamic model.

Extensive simulations were performed and the results demonstrate that, unlike the classical linear model for flexible robots whose tracking of its corresponding nonlinear model falls outside the acceptable limits, the proposed linear model presents excellent tracking of its corresponding nonlinear detailed model. The results show that the proposed linear model achieves steady-state behavior within 0.5 seconds of the time its

corresponding nonlinear model achieves steady state, and its steady-state behavior is within 2° of its corresponding nonlinear model steady-state behavior.

This chapter also presented a solution to the problem of simulating velocity behavior, when taking into account the effects of coulomb or stick-slip friction, with Runge-Kutta integration algorithms. Without the proposed modification, this type of simulation leads to a velocity time response full of erroneous high frequency oscillations which do not diminish with time. With the proposed modification, the velocity time response is accurately modeled - the erroneous high frequency oscillations are dampened out while the true velocity oscillations are not.

Chapter 3: CONTROLLER DESIGN

3.1 INTRODUCTION

As the literature survey indicated, the most promising robot controllers are developed with nonlinear techniques; however, these controllers are highly mathematical, have their own limitations, and their implementation is complex. Linear controllers that have fewer limitations can be designed and are most often used in industrial applications. Linear controllers are limited by the accuracy of the linear model. The linear models proposed in the past for flexible-joint robots only track the nonlinear (true) model in regions near the linearization point. In Chapter 2, a linear model for flexible-joint robots was derived, that provides accurate tracking of the corresponding nonlinear model over the entire variable range. This allows the derivation of a linear controller which does not suffer from the limitations of the nonlinear controllers. This derivation is the second objective of this thesis and is presented in this chapter.

In Chapter 2, it was understood that the proposed linear model is not an exact match for the nonlinear model. In order to account for this difference between the linear model and the nonlinear model, a stochastic-error term was derived and added to the linear model. During the control phase, this error term acts as a plant disturbance. In Chapter 2, it was also indicated that, in general, exact measurement of all states is expensive and is often not feasible. To overcome this handicap, the proposed controller will assume that only the position variables can be measured, and that this measurement is corrupted by noise.

Because of the uncertainty existent in this resulting linear model, robust control theory is the most suitable technique, allowing robust performance and robust stability to be achieved. One of the existing controller design approaches which is known to achieve the desired performance with minimum control effort is the Linear Quadratic Gaussian / Loop Transfer Recovery (LQG/LTR) approach. This method is used in this chapter to design a robust position-tracking controller for the study-case flexible-joint robot. After its derivation, the robustness and position-tracking abilities of this controller are demonstrated in several simulations with various reference trajectories.

The remainder of this chapter is organized as follows. Section 3.2 contains an overview of LQG/LTR control methodology. Section 3.3 presents the design of the proposed LQG/LTR controller for the study-case robot. Section 3.4 contains the simulations which demonstrate the robustness and position-tracking capabilities of the proposed controller. Finally, Section 3.5 contains some conclusions.

3.2 OVERVIEW OF LQG/LTR CONTROLLER METHODOLOGY

This section provides an overview of the methodology of LQG/LTR controllers. The actual design of the proposed controller is described in Section 3.3.

3.2.1 INTRODUCTORY CONCEPTS REGARDING OPTIMAL ROBUST CONTROL

The Plant Model

For optimal robust linear control theory, the plant is described by a state-space equation in the form of

$$\dot{\mathbf{X}} = \mathbf{A}\,\mathbf{X} + \mathbf{B}\,\mathbf{U} + \mathbf{G}\,\mathbf{w} \qquad \mathbf{Y} = \mathbf{C}\,\mathbf{X} + \mathbf{v} \tag{3-1}$$

where **w** and **v** are zero-mean Gaussian stochastic processes, uncorrelated in time, and uncorrelated with each other, with covariances

$$\mathbf{E\{ww^T\} = W \geq 0}, \qquad \mathbf{E\{vv^T\} = V > 0}, \qquad \mathbf{E\{wv^T\} = 0} \tag{3-2}$$

The term **Gw** in Equation (3.1) describes the plant disturbance, while the term **v** is the plant measurement noise.

The Singular Gains

In multivariable linear controller design, the study of the singular gains replaces the study of other system characteristics often employed in linear control, such as Bode or Nyquist diagrams. The singular gains of a given gain matrix **T(jw)** are defined as

$$\sigma_i = \sqrt{\text{eigenvalue}_i \text{ of } \{\mathbf{T^H(jw)T(jw)}\}}\ , \text{ i=1..total \# of eigenvalues of } \mathbf{T(jw)} \tag{3-3}$$

Theory indicates that the frequency characteristics of a multivariable Multi-Input Multi-Output (MIMO) system are sandwiched between the frequency characteristics of the smallest and the largest singular gains of the system.

The Return Ratios

For a simple closed loop system with forward-path transfer function G(s) and return-loop transfer function H(s), the product G(s)H(s) is named the "return ratio". In general, the return ratio refers to a gain product, and depends on what is being considered the input and the output node for a given block diagram. The singular gains of various return ratios are inspected during the design of robust controllers.

3.2.2 LQG/LTR CONTROL THEORY

The LQG/LTR controller for the state-space model (3-1) is designed in two steps, outlined in simple, summarized form in Figure (3-1). The remainder of this section describes each of these two steps in more detail. Figure (3-2) outlines the operation of the controlled plant, also in simple form.

The first step consists of the design of a deterministic full-state optimal Linear Quadratic feedback Regulator/controller (LQR). This LQR is deterministic in that it assumes an uncertainty (disturbance and noise) -free plant model, of the form

$$\dot{\mathbf{X}} = \mathbf{A}\,\mathbf{X} + \mathbf{B}\,\mathbf{U} \qquad \mathbf{Y} = \mathbf{C}\,\mathbf{X}. \tag{3-4}$$

This LQR is a full-state regulator/controller in that it assumes a value for **C** of identity in Equation (3-4); i.e. that all states are available for feedback. This LQR is optimal and quadratic in that, it is designed such that the following quadratic performance index is minimized:

$$\mathbf{J_1} = \lim_{T\to\infty} \mathbf{E}\left\{\int_0^T (\mathbf{X^T Q X} + \mathbf{U^T R U})dt\right\} \tag{3-5}$$

In (3-5), **E** is the expectation operator, the quadratic term $\mathbf{X^T Q X}$ describes the energy of the tracking error, the quadratic term $\mathbf{U^T R U}$ describes the energy of the input, and **Q** and **R** are weighing matrices. The control law that minimized the performance index (3-5) is given by the feedback law

$$\mathbf{U} = -\,\mathbf{K_C}\,(\mathbf{X}-\mathbf{X_{REF}}) \tag{3-6}$$

where $\mathbf{X_{REF}}$ is the reference trajectory, and $\mathbf{K_C}$ is the controller gain. $\mathbf{K_C}$ is given by

$$\mathbf{K_C} = \mathbf{R^{-1}}\,\mathbf{B^T}\,\mathbf{P_C} \tag{3-7}$$

where $\mathbf{P_C}$ is the unique solution of the Ricatti equation

$$\mathbf{A}^{\mathbf{T}}\mathbf{P}_{\mathbf{C}} + \mathbf{P}_{\mathbf{C}}\mathbf{A} - \mathbf{P}_{\mathbf{C}}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{P}_{\mathbf{C}} + \mathbf{Q} = 0 \tag{3-8}$$

In this thesis, the solution to Equation (3-7) and (3-8) is obtained with the MATLAB™ function LQR(A,G,C,W,V).

It is known that the actual plant, described by Equations (3-1) contains uncertainties, does not use the feedback of all states, and the available feedbacks are corrupted by noise. In order to take into account all this, and to provide an estimate $\hat{\mathbf{X}}$ of the full-state vector, a Linear Quadratic Estimator/observer (LQE) is designed next. This observer is designed such that the estimation of the error between the estimated state and the true state is minimized. This minimization goal is expressed in equation form as,

$$\mathbf{J}_2 = \lim_{T\to\infty} \mathbf{E}\left\{\int_0^T (\mathbf{X} - \hat{\mathbf{X}})^{\mathbf{T}}(\mathbf{X} - \hat{\mathbf{X}})dt\right\} \tag{3-9}$$

The observer which achieves this is a Kalman Filter and its block diagram is given by Figure (3-3). This observer uses $\mathbf{Y}$ and $\mathbf{U}$ as inputs, and outputs the state estimate, $\hat{\mathbf{X}}$, which is then used as the an exact value for the full state $\mathbf{X}$ by the LQR. The gain matrix K_F of the observer is given by

$$\mathbf{K}_{\mathbf{F}} = \mathbf{P}_{\mathbf{F}}\mathbf{C}^{\mathbf{T}}\mathbf{V}_{\mathbf{F}}^{-1}, \quad \mathbf{V}_{\mathbf{F}} > \mathbf{0} \tag{3-10}$$

where $\mathbf{P}_F$ is the unique solution of the Ricatti equation

$$\mathbf{P}_{\mathbf{F}}\mathbf{A}^{\mathbf{T}} + \mathbf{A}\mathbf{P}_{\mathbf{F}} - \mathbf{P}_{\mathbf{F}}\mathbf{C}^{\mathbf{T}}\mathbf{V}_{\mathbf{F}}^{-1}\mathbf{C}\mathbf{P}_{\mathbf{F}} + \mathbf{G}_{\mathbf{F}}\mathbf{W}_{\mathbf{F}}\mathbf{G}_{\mathbf{F}}^{\mathbf{T}} = 0 \tag{3-11}$$

The best observer performance is obtained using

$$\mathbf{V}_{\mathbf{F}} = \mathbf{V}, \quad \mathbf{W}_{\mathbf{F}} = \mathbf{W}, \text{ and } \quad \mathbf{G}_{\mathbf{F}} = \mathbf{G}, \tag{3-12}$$

in Equations (3-10)-(3-11), where $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{G}$ were given by Equations (3-1)-(3-2).

The final controller for the plant model (3-1) is obtained by cascading the LQR with the LQE, as shown in simple form in Figure (3-2) and in more detail in Figure (3-4). This final controller is referred to a Linear Quadratic Gaussian (LQG) controller. However, it has been shown that the performance achieved by the LQR is lost when the LQR is coupled with the LQE. In order to maintain the desired performance achieved by the LQR, Loop Transfer Recovery (LTR) techniques should be used to design the LQE. As part of the LTR approach, matrices $\mathbf{V_F}$, $\mathbf{W_F}$, and $\mathbf{G_F}$ are no longer prescribed by (3-1) and (3-2) and become the design parameters of the observer. In the proposed research, the solution to Equations (3-10) and (3-11) is obtained with the MATLAB^{TM} function LQE(A,G,C,W,V).

According to Figure (3-4), the return ratio at node #1, $\mathbf{L_1}$, is given by

$$\mathbf{L_1} = -\mathbf{K_C}\,(s\mathbf{I}-\mathbf{A})^{-1}\,\mathbf{B} \tag{3-13}$$

while the return ratio at node #2 is given by

$$\mathbf{L_2} = \mathbf{K_C}\,[(s\mathbf{I}-\mathbf{A}) + \mathbf{K_F}\,\mathbf{C}+\mathbf{B}\,\mathbf{K_C}]^{-1}\,\mathbf{K_F}\,\mathbf{C}\,(s\mathbf{I}-\mathbf{A})^{-1}\,\mathbf{B} \tag{3-14}$$

During the design of the LQR, the weighing matrices $\mathbf{Q}$ and $\mathbf{R}$ are varied while the system's time response, and the behavior of the singular gains of the return ratio $\mathbf{L_1}$, are monitored. The systems' time response is monitored to assure that the system provides accurate tracking of the reference signal. The behavior of the singular gains of $\mathbf{L_1}$ are monitored to assure that the cross-over frequencies of this return ratio are sufficiently close to the system's characteristic frequency. Below this characteristic frequency, the plant disturbances are active and the controller must have sufficiently high gains to be able to reject them. At values above this characteristic frequency, measurement noise becomes active, and the controller gain at these frequencies must be almost non-existent

to minimize its effect. During the design of the LQE, the parameters $\mathbf{V_F}$, $\mathbf{W_F}$ and $\mathbf{G_F}$ are varied while the system's time response, and the behavior of the singular gains of the return ratio $\mathbf{L_2}$, are monitored. The system's time response is monitored to assure continued acceptable reference-signal tracking. The singular gains of $\mathbf{L_2}$ are monitored to assure that they are sufficiently close to the singular gains of $\mathbf{L_1}$, in order to recover the robustness properties achieved by the LQR.

These, and other details regarding LQG/LTR control, can be found in any multivariable linear control textbooks, e.g. Maciejowksi, '89 [20].

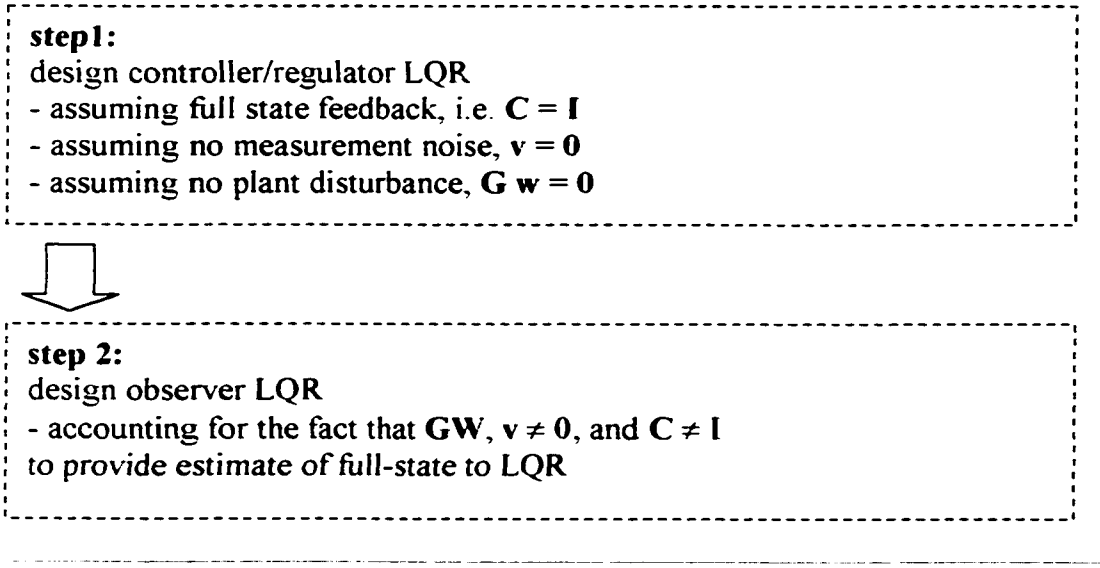


Figure 3-1: The Two Steps in Designing a LQG/LTR Controller for the Plant Model $\dot{X} = A X + B U + G w$, $Y = C X + v$

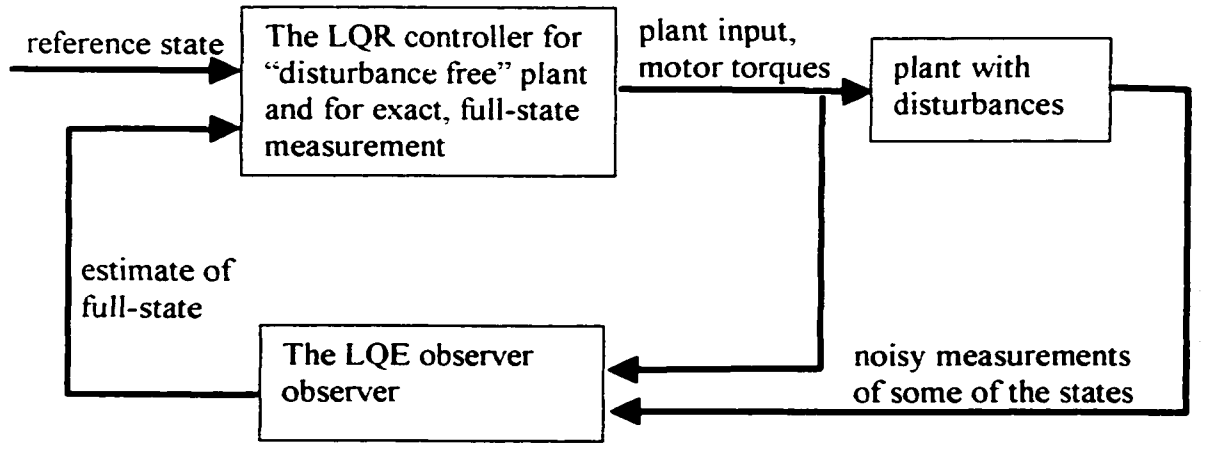


Figure 3-2: Simplified Description of LQG/LTR Controlled Plant

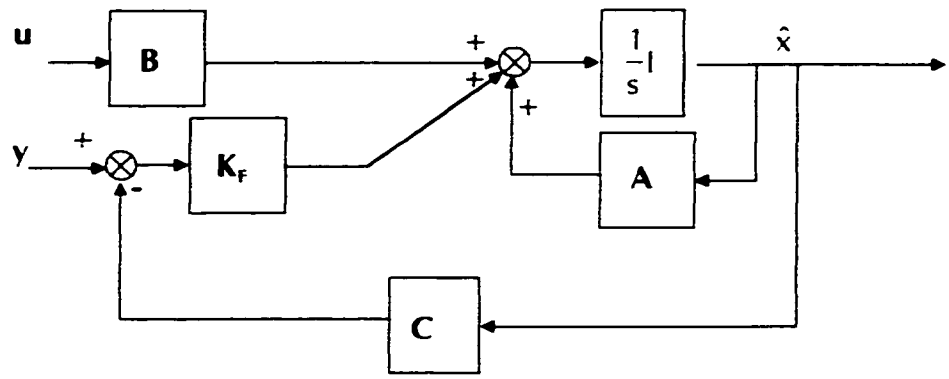


Figure 3-3: Linear Optimal Observer: Kalman Filter

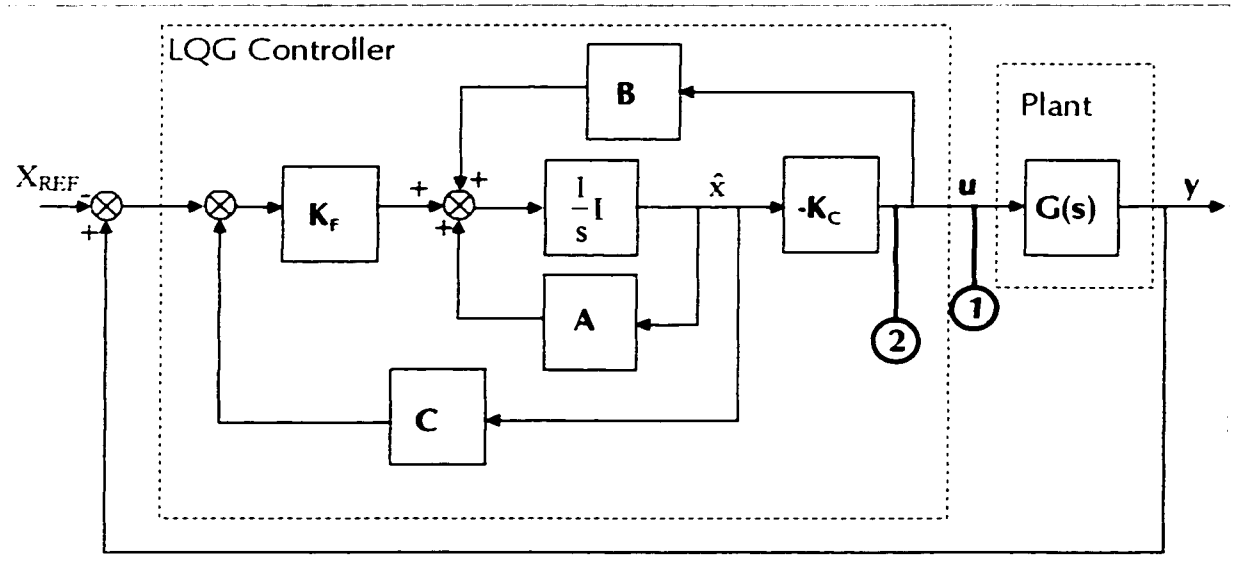


Figure 3-4: The Overall Compensated System

3.3 STUDY-CASE LQG/LTR CONTROLLER DESIGN

This section describes the design of the proposed motion-tracking controller for the study-case flexible-joint robot introduced in Chapter 1. Section 3.3.1 introduces the pertaining study-case model parameters not discussed thus far. The LQR design is described in Section 3.3.2, while the LQE design is described in Section 3.3.3. The resulting controller parameters, gains $\mathbf{K_F}$ and $\mathbf{K_C}$ of Figure (3-4) are presented in Section 3.3.4. The validation of this controller is presented next, in Section 3.4.

3.3.1 MODEL PARAMETERS

The model for Equation (3-1) for the general n-link flexible-joint robot case was derived in Chapter 2. The specific form of this equation for the study-case robot was derived in Appendix 1, and values for matrices **A**, **B**, **G**, and **W** for the study-case robot were given in Section 2.5. In Equation (3-1), the output matrix **C** is chosen as

$$\mathbf{C} = \text{diag}(1,\ 1,\ 0,\ 0,\ 1,\ 1,\ 0,\ 0) \tag{3-15}$$

signifying that only the positions of the motors and the links are used for feedback. This considerably reduces the cost of the controller design, which makes the proposed method very attractive. For certainty, the measurement noise is known to be at least one half of the sensors' resolutions. However, there may be other noise sources. Therefore, to be on the safe side, the measurement noise is taken to be five times larger than the sensors' resolutions, and is given by Equation (3-16). In Equation (3-16), N(0,1) denotes a Gaussian white noise, with zero mean and unit variance.

$$\mathbf{v} = \begin{bmatrix} 5 \times \text{resolution of motor1 position sensor} \times N(0,1) \\ 5 \times \text{resolution of motor2 position sensor} \times N(0,1) \\ 0 \\ 0 \\ 5 \times \text{resolution of link1 position sensor} \times N(0,1) \\ 5 \times \text{resolution of link2 position sensor} \times N(0,1) \\ 0 \\ 0 \end{bmatrix}, \tag{3-16}$$

The characteristic frequency of the study-case robot system is 35 rad/s. The maximum torques that the motor #1 and motor #2 can provide are 100 N.m and 30 N.m, respectively.

3.3.2 LQR DESIGN

The LQR design consists of deriving the values for the **Q** and **R** weighing matrices which result in the desired performance. This is generally a trial-and-error process. Mathematical theory presents a few techniques for manipulating elements of the **Q** and **R** matrices in order to shape the singular gains of the system, but these techniques do not work very well for all cases.

In the work of Hsiao et al '96 [14], an iterative technique for optimizing the LQR parameters is derived and used. In the research work of this thesis, a similar iterative technique was employed but it was rather based on random searches and it was not fully automatic. For the current application, the LQR parameters **Q** and **R** are 8x8 and 2x2 matrices, respectively. Assuming diagonal form for both **Q** and **R**, there still are a total of ten parameters that can vary in a linear fashion, from 10^{-5} to approximately 1000. As a result, it was considered that a linear search would require too long a computation time, and a random search was used instead. Each random search was run for 20,000 cycles.

The search was carried out with the goal of (1) minimizing the difference between the tracking signal and the plant output in the time response, and (2) achieving a cross-over frequency for the singular gains as close as possible to the robot system characteristic frequency. For a given variable range, the random search was carried out six times. This allowed a human operator to observe which sub-range of values for each of the ten parameters tended to give the best results, and focused in on those sub-ranges in subsequent searches. The search was carried out on the deterministic plant model given by (3.2). When an LQR matrix gain was found which resulted in the best achievable results, the search was stopped and the LQE was then designed. The LQE design process is described in Section 3.3.4. The resulting LQR matrix gain, $\mathbf{K_C}$ is given in Section 3.3.5.

3.3.4 LQE DESIGN WITH LTR TECHNIQUES

The LQE design consists of deriving the values for the parameters $\mathbf{W_F}$, $\mathbf{V_F}$ and $\mathbf{G_F}$. LTR theory indicates that to achieve the desired LQE goals, the LQE parameter $\mathbf{W_F}$ must be of the form

$$\mathbf{W_F} = q\,\mathbf{G_F}\,\mathbf{G_F}^T + q^2\,\mathbf{B}\,\mathbf{V_F}\,\mathbf{B}^T, \qquad (3\text{-}17)$$

where q is a scalar and becomes another design parameter. LTR theory also suggests several methodologies that can be used to design the LQE parameters $\mathbf{G_F}$ and $\mathbf{V_F}$ but these are generally very mathematical and are not always applicable. In this thesis, the design of parameters q, $\mathbf{G_F}$ and $\mathbf{V_F}$ was conducted through a search similar to that used for the design of the LQR parameters. This search was carried out for the full plant model, given by (3-1). The search goals were (1) to minimize the tracking error in the

time response simulations, (2) to minimize the difference between the singular gains of return ratio $\mathbf{L}_2$ and the singular gains of return ratio $\mathbf{L}_1$, and (3) to assure that the control torques necessary were below the saturation limits of the motors. Goals (1) and (2) conflict with goal (3) as better controller performance require higher controller input torque. However, if the required controller input torque is more than the motors can provide, a saturation occurs. This saturation is a nonlinear dynamic event, and the linear dynamic equations do not provide adequate modeling of the system behavior under saturation conditions. The resulting value for the LQE gain matrix $\mathbf{K_F}$ is given in the next section.

3.3.5 FINAL PROPOSED CONTROLLER

In summary, the proposed controller is described in Figure (3-4), where the two gain matrices $\mathbf{K_C}$ and $\mathbf{K_F}$ are given by Equations (3-18) and (3-19), respectively:

$$\begin{bmatrix} 510.16 & -53.891 & 3.8969 & 1.0415 & 1519.8 & -85.10 & 148.31 & 8.7826 \\ -110.32 & 624.79 & -0.68747 & 0.62997 & 249.32 & 1405.2 & 29.945 & 31.12 \end{bmatrix} \quad (3\text{-}18)$$

$$\begin{bmatrix} 221.59 & -0.0836 & 0 & 0 & 0.00064038 & -0.0045668 & 0 & 0 \\ -0.0836 & 231.16 & 0 & 0 & -0.0020869 & 0.062751 & 0 & 0 \\ 24.551 & -9.7771 & 0 & 0 & 0.043191 & 1.1085 & 0 & 0 \\ -28.103 & 2.6717 & 0 & 0 & -0.21638 & 4.8994 & 0 & 0 \\ 0.64047 & -2.0873 & 0 & 0 & 0.013482 & 0.025162 & 0 & 0 \\ -0.45673 & 6.2757 & 0 & 0 & 0.0025161 & 0.09545 & 0 & 0 \\ 98.898 & -265.99 & 0 & 0 & 0.0020057 & -0.28174 & 0 & 0 \\ -212.64 & 961.32 & 0 & 0 & 0.015058 & 0.19754 & 0 & 0 \end{bmatrix} \quad (3\text{-}19)$$

3.4 CONTROLLER VALIDATION SIMULATIONS

In this section, results which demonstrate that the proposed motion-tracking controller meets the desired criteria are presented. The first figure, Figure (3-5) presents the resulting singular gains of the two returned ratios monitored, $\mathbf{L_1}$ and $\mathbf{L_2}$. Figures (3-6) to (3-10) contain the resulting link positions in several time responses of the controlled plant with different reference trajectories. These reference trajectories and their corresponding figures are described in Table (3-1).

trajectory #	main trajectory features (all end in steady-state)	results shown in
#1	steep (high speed) ramp, average corners	Figure (3-6)
#2	4^{th} order polynomial, smooth corners, low speed	Figure (3-7)
#3	sinusoidal, end effector traces a continuos circle; no corners; links move in opposite directions	Figure (3-8)
#4	4^{th} order polynomial followed by ramp; sharp corners; links move in opposite directions	Figure (3-9)

Table 3-1: Descriptions of Position-Tracking Time Response Simulations

As can be seen, the trajectories were chosen to test the controller performance under a variety of conditions.

Because the current application is motion tracking, it is the links' positions that are controlled to track the reference trajectory. Due to the nature of flexible-joint dynamics, the motors' positions are not themselves controlled to always track the reference trajectory but rather, they are controlled to cause the links' positions to provide the correct tracking. As a result, only the links' positions are inspected as indication of the position-tracking capability of the controller. The resulting links' and motors'

velocities are also not tracked because (1) they are not specified in this type of application and (2) safe values for these velocities are automatically dictated by the limitations imposed on the required motor torques.

Discussion of Controller Frequency Behavior

Figure (3-5) shows the singular gains of the returned ratios monitored, namely L_1 and L_2. This figure indicates that the returned ratio at the plant output, L_2 is sufficiently close to the returned ratio at the plant input, L_1, for frequencies less than 200 rad/s. This will guarantee that the robustness properties are recovered at the plant output. After the 200 rad/s, the returned ratio at the plant output drops considerably. This results in lower values at the higher frequencies, which is actually beneficial because it indicates increased resistance to measurement noise which becomes active at high frequencies. The singular gains of the controlled plant are also shown to have a cross-over frequency in the neighborhood of the flexible-joint robot's characteristic frequency (indicated by construction lines in Figure (3-5)), as desired.

Discussion of Controller Path-Tracking and Disturbance Rejection Capability

Resulting path-tracking errors and maximum required motor torques for the four trajectories tested are presented in Table (3-2). The results indicate that at times the percentage position errors seem high, even though acceptable. However, the absolute position errors observed in these results are fairly regular, around 1.2° for link 1 and around 0.5° for link 2, despite the fact that the simulations were run for different speeds. This indicates that the position errors are a direct result of the disturbance and noise added, which also do not depend on the speeds.

reference trajectory	maximum position error		maximum motor torque	
	link 1	link 2	motor 1	motor 2
#1	1.5° = 1.5%	0.2° = 0.2%	75N.m	27N.m
#2	1° = 10%	0.4° = 4.0%	35N.m	25N.m
#3	1.4° = 4.7%	0.4°=1.4%	57N.m	27N.m
#4	0.9° = 6%	0.8%=6%	83N.m	27N.m

Table 3-2: Results of Controller Time Response Simulations

The disturbance and noise added are of considerable magnitude (refer to Section 2.5) and are continously added directly to the plant model, at each time step. As a result, the controller cannot and is not expected to diminish their effect over time and eventually nullify it, but rather to dampen/diminish it at each time step. This behavior is apparent in the four time simulations, as the controlled links' positions continue to exhibit small oscillations about the reference positions. However, these oscillations are very small in value, and, therefore, these simulations indicate the controller provides very good disturbance rejection. This achievement is further enhanced by the fact that the only feedbacks used are noise-corrupted position measurements.

This constitutes the validation of the proposed controller.

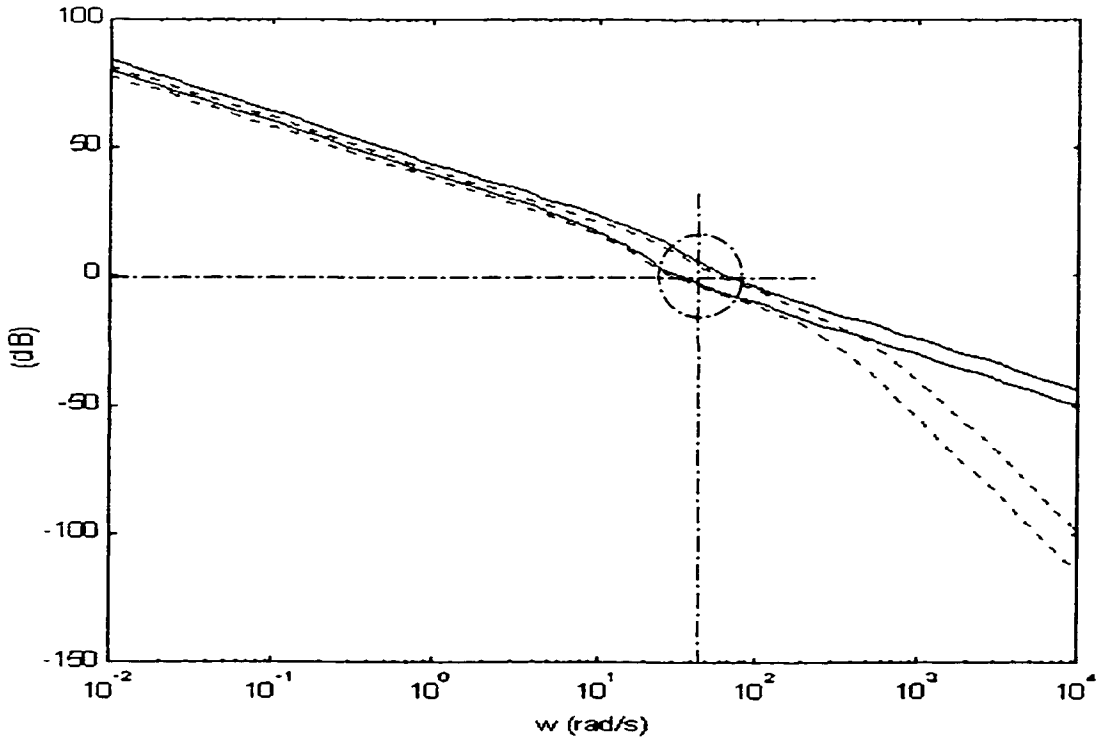
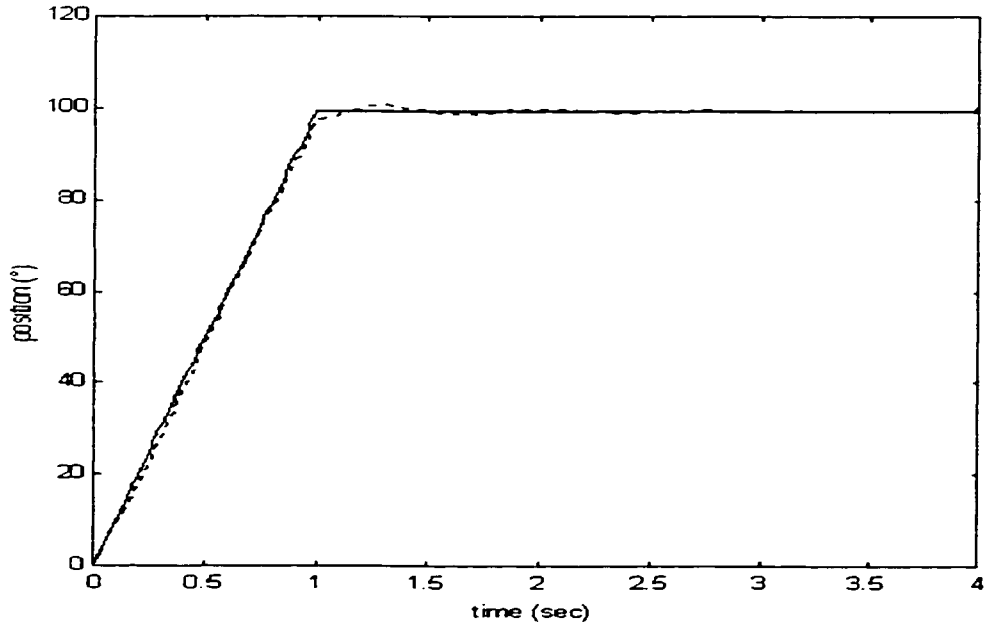


Figure 3-5: Singular Gains of Monitored Return Ratios L_1 (solid), L_2 (dotted)

(a) Link 1 Position



(b) Link 2 Position

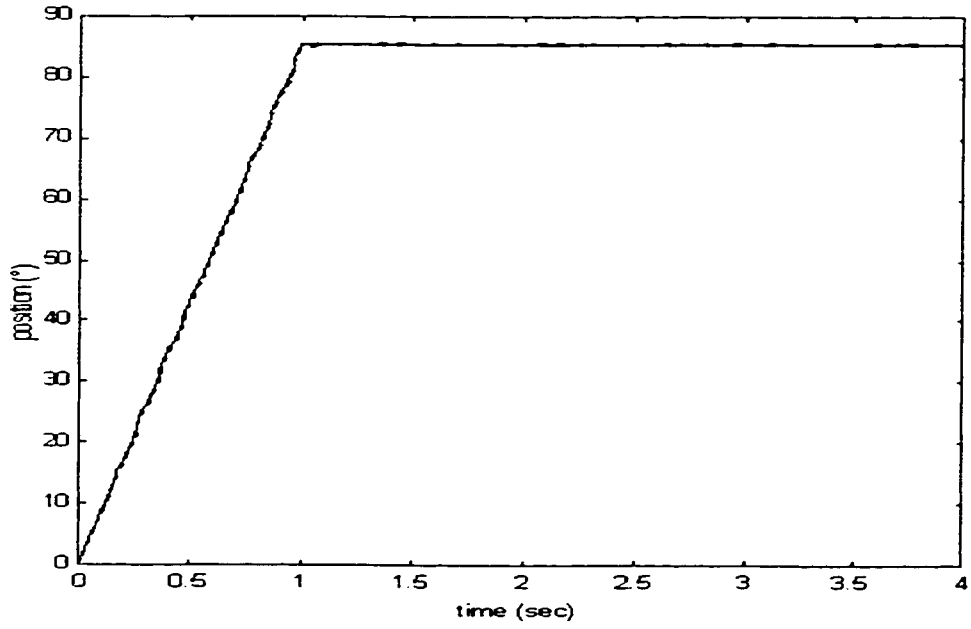
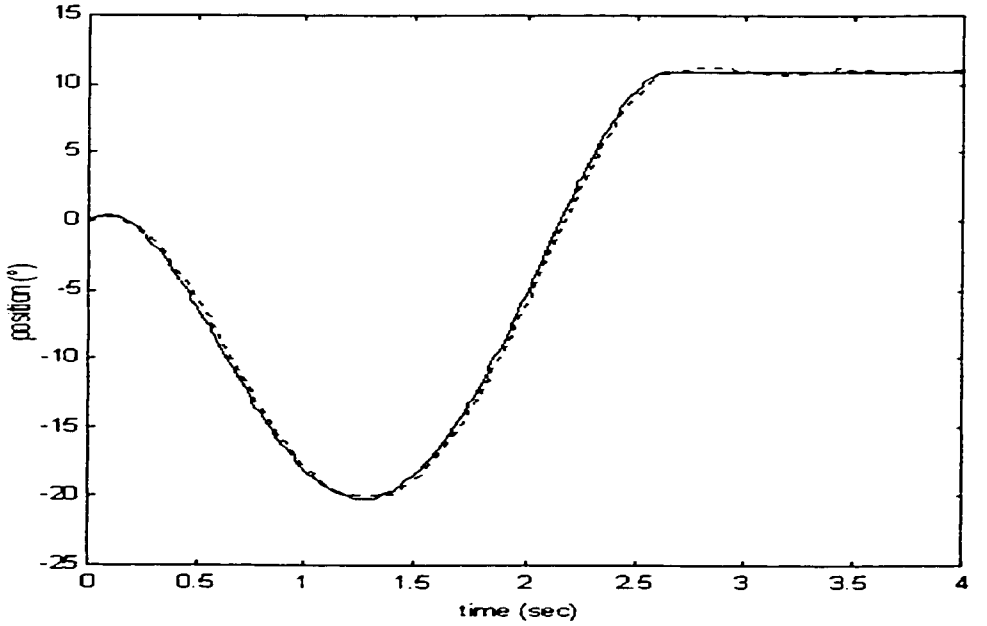


Figure 3-6: Time Response Simulation with Reference Trajectory #1
(solid) reference trajectory, (dotted) actual achieved trajectory

(a) Link 1 Position



(b) Link 2 Position

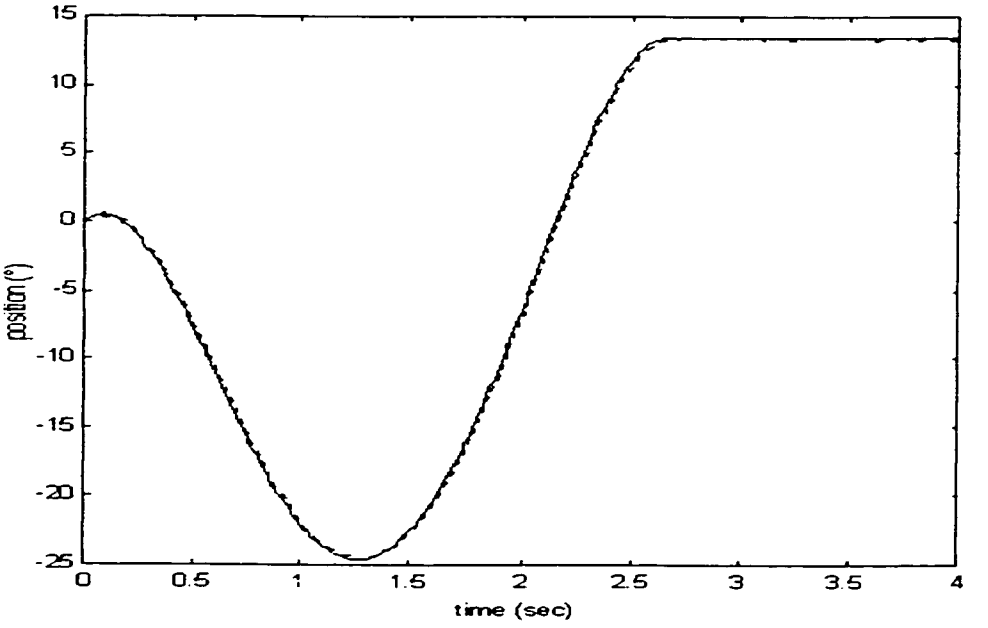
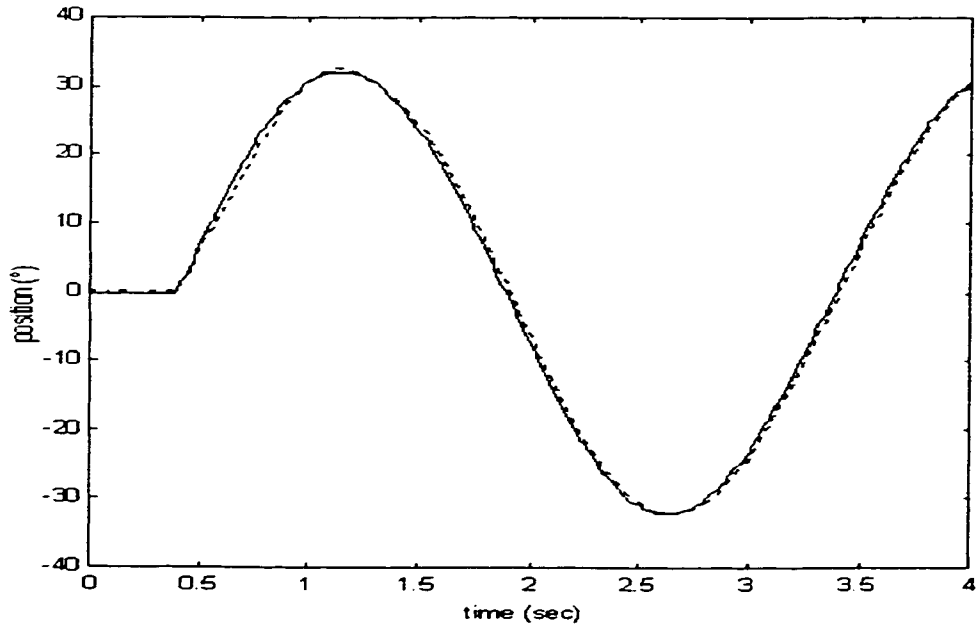


Figure 3-7: Time Response Simulation with Reference Trajectory #2
(solid) reference trajectory, (dotted) actual achieved trajectory

(a) Link 1 Position



(b) Link 2 Position

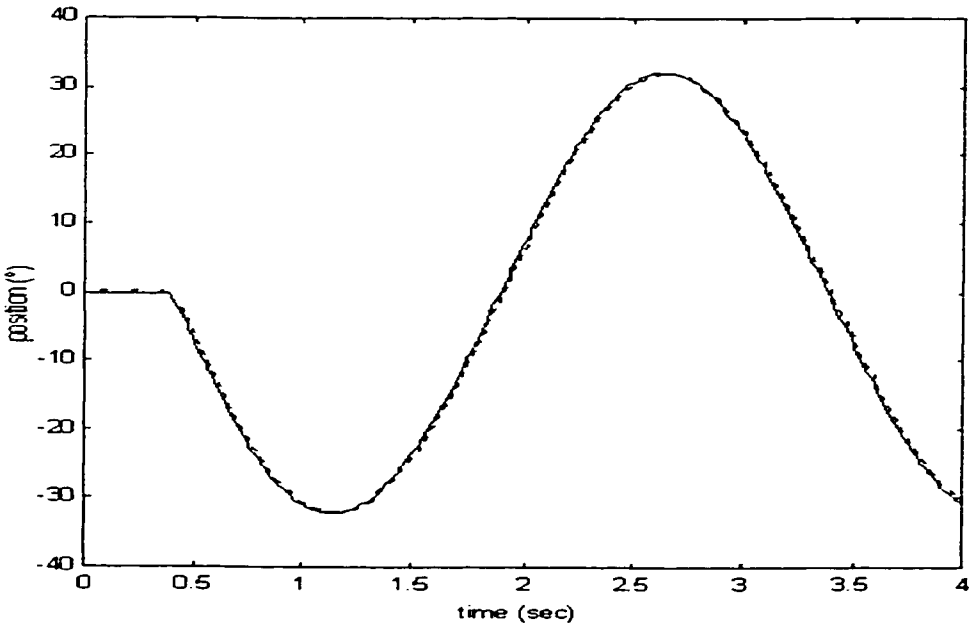
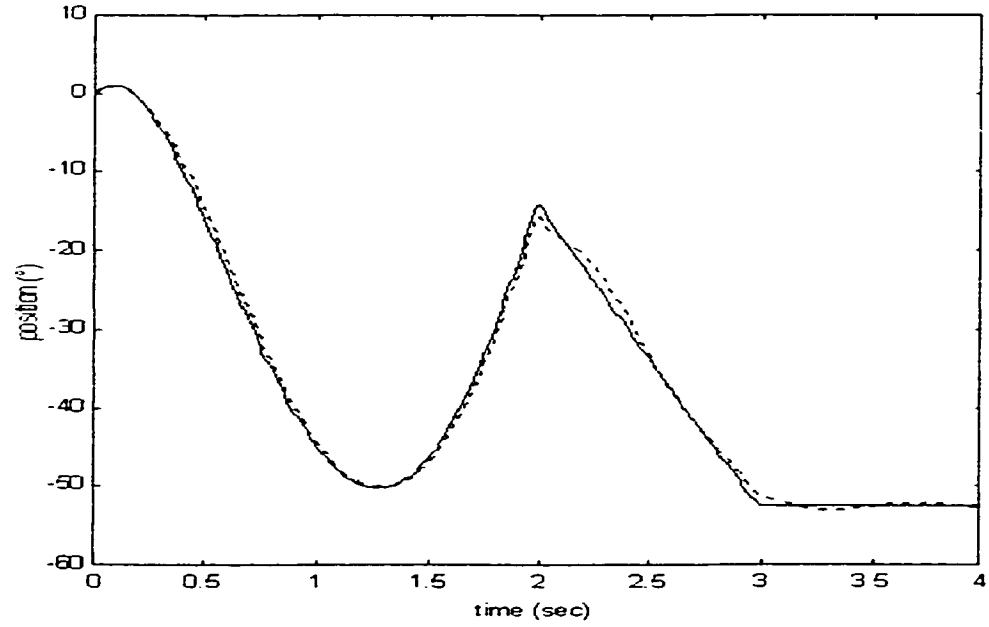


Figure 3-8: Time Response Simulation with Reference Trajectory #3
(solid) reference trajectory, (dotted) actual achieved trajectory

(a) Link 1 Position



(b) Link 2 Position

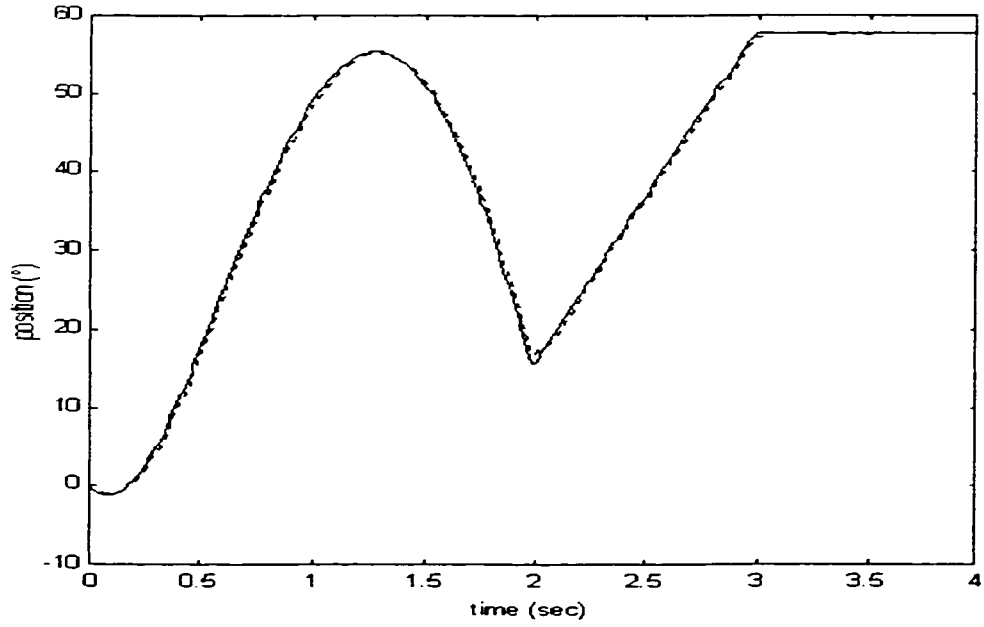


Figure 3-9: Time Response Simulation with Reference Trajectory #4
(solid) reference trajectory, (dotted) actual achieved trajectory

3.5 CONCLUSIONS

This chapter presented the derivation of a robust optimal linear motion-tracking controller for the linear dynamic model of the flexible-joint robots presented in Chapter 2. The design of this controller was based on linear quadratic Gaussian (LQG) techniques coupled with loop transfer recovery (LTR) techniques. This controller used only position-sensor feedback. After its development, the derived controller was tested in a series of time response simulations with different reference trajectories, for a plant model which was artificially corrupted by plant disturbances and measurement noise. The simulations results indicate that the controller provides good reference-trajectory tracking, despite the high level of plant disturbances and measurement noise introduced. All torques necessary for the motion-tracking control were below the motors' saturation limits.

Chapter 4: CONTROLLER EXPERIMENTAL IMPLEMENTATION

4.1 INTRODUCTION

The final test of the controller derived in this thesis is an experimental implementation on an actual flexible-joint robot. This chapter describes the experimental setup for this implementation, and the derivation of the algorithmic form of the proposed controller in anticipation of this implementation. The actual implementation of this controller algorithm will be carried out in future work.

This chapter is organized as follows. Section 4.2 presents the relevant details of the experimental flexible-joint robot setup. Section 4.3 describes the software aspect of the controller implementation. Section 4.4 contains some conclusions.

4.2 EXPERIMENTAL SETUP

A schematic diagram of the experimental setup is presented in Figure (4-1). To facilitate discussion, this setup was subdivided into three elements: (1) the robot setup, (2) the controller, and (3) the PC. Each of these elements will be discussed in detail in the next three sub-sections.

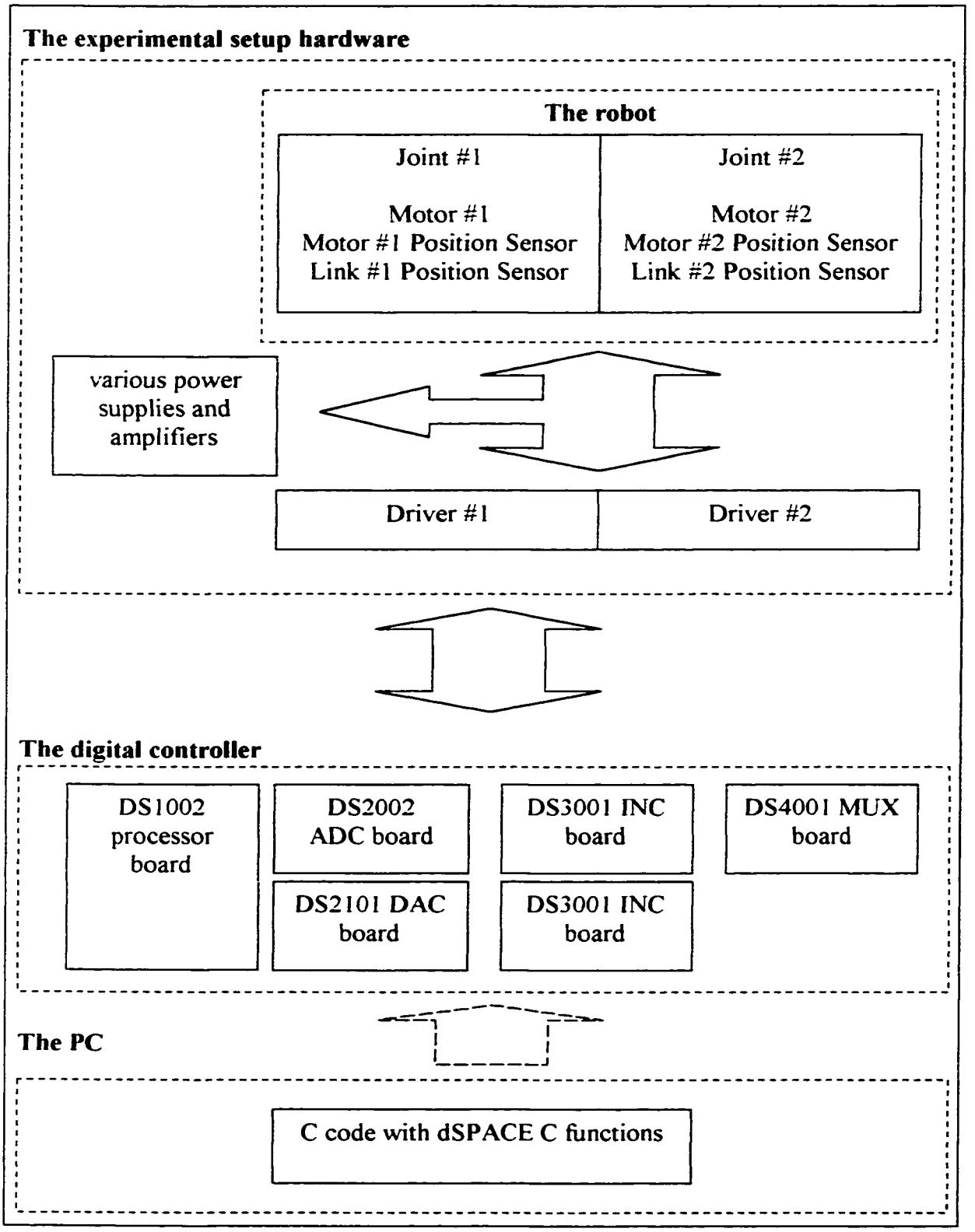


Figure 4-1: Experimental Setup for Controller Test Implementation

4.2.1 ROBOT SETUP

The flexible-joint robot used for the proposed controller implementation was introduced in Chapter 1, and is a two-link planar revolute-joints robot. This robot was designed and built in the Flexible Manufacturing Centre at McMaster University (ElMaraghy et al '94 [10], Massoud '94 [21]) and has been used in several research projects sponsored by the Institute for Robotics and Intelligent Systems (IRIS). This robot is now in the Intelligent Manufacturing Systems (IMS) Centre at the University of Windsor.

In order to emphasize the joint flexibility problem in this robot, the shafts that normally connect the motors to the links have been replaced by pairs of counter-set helical springs. If the shafts were cylindrical and made out of steel, the spring constants of joint 1 and joint 2 would have been approximately 89 kN.m and 56 kN.m, respectively. The helical-spring shafts have spring constants of 137 N.m and 100 N.m, respectively. The two motors are outer-rotor direct-drive and hence have no connecting gears. This implies that the joints present no backlash. The motors can be controlled in several modes. In the current torque-control mode, the generated torque is directly proportional to the command voltage. There are four position sensors, one built in each motor, and one placed on each link. The numerical details of the motors and of the sensors are given in Table (4-1) and Table (4-2), respectively. The system is controlled by two DYNASERV^{TM} drivers. There are several power boxes and amplifiers that are also part of the hardware setup of the robot manipulator. The dimensional information of the robot is provided in Appendix 1.

motor	Peak Torque (N.m)	Rated Speed (rev /sec)
motor 1	100.0	1.0
motor 2	30.0	2.0

Table 4-1: Robot Motor Specifications

Position Sensor Location	Sensor pulses/rev
motor 1	1024000
motor 2	655360
link 1	32000
link 2	32000

Table 4-2: Resolution of the Robot Position Sensors

4.2.2 CONTROLLER

The controller consists of several dSPACE^{TM} electric boards. These boards and their functions are described in Table 4.3.

board	description and function
DS1002	processing board with floating-point Digital Signal Processor (DSP) • Interfaces with PC and other boards • Performs digital processing • Contains memory storage for variables • Provides watchdog and I/O supervision
DS2002	A/D Conversion board • not used for this application
DS2101	D/A Conversion board • Used to send voltage signal to drivers to generate required motor torques
DS3001	Incremental Encoder board • Used to read all position sensors
DS4001	Timer Digital Multi I/O board • Oversees all data transfers between the boards and check • Checks status of robot components, providing further safety checks

Table 4-3: Digital Controller Boards

4.2.2 THE PERSONAL COMPUTER

The user interacts with the PC using several pieces of software provided by dSPACE™[8]. For the purposes of this research, the controller real-time code for the experimental robot is written in C using several libraries from dSPACE™. Upon completion, the C code is compiled and downloaded into the DS1002 processing board. From there, the controller code can be run on-line, controlling the robot, or off-line, with the power to the robot drivers shut off. During the on-line run, the PC user can monitor some of the controller variables and can issue immediate stop-operation orders to the robot. All the desired controller variables can be saved but may not be available for real-time observation.

4.3 SOFTWARE IMPLEMENTATION

Figure (4-2) contains the controller action in a flowchart form. Each of the seven steps appearing in this flowchart is further detailed in Figures (4-3) to (4-10).

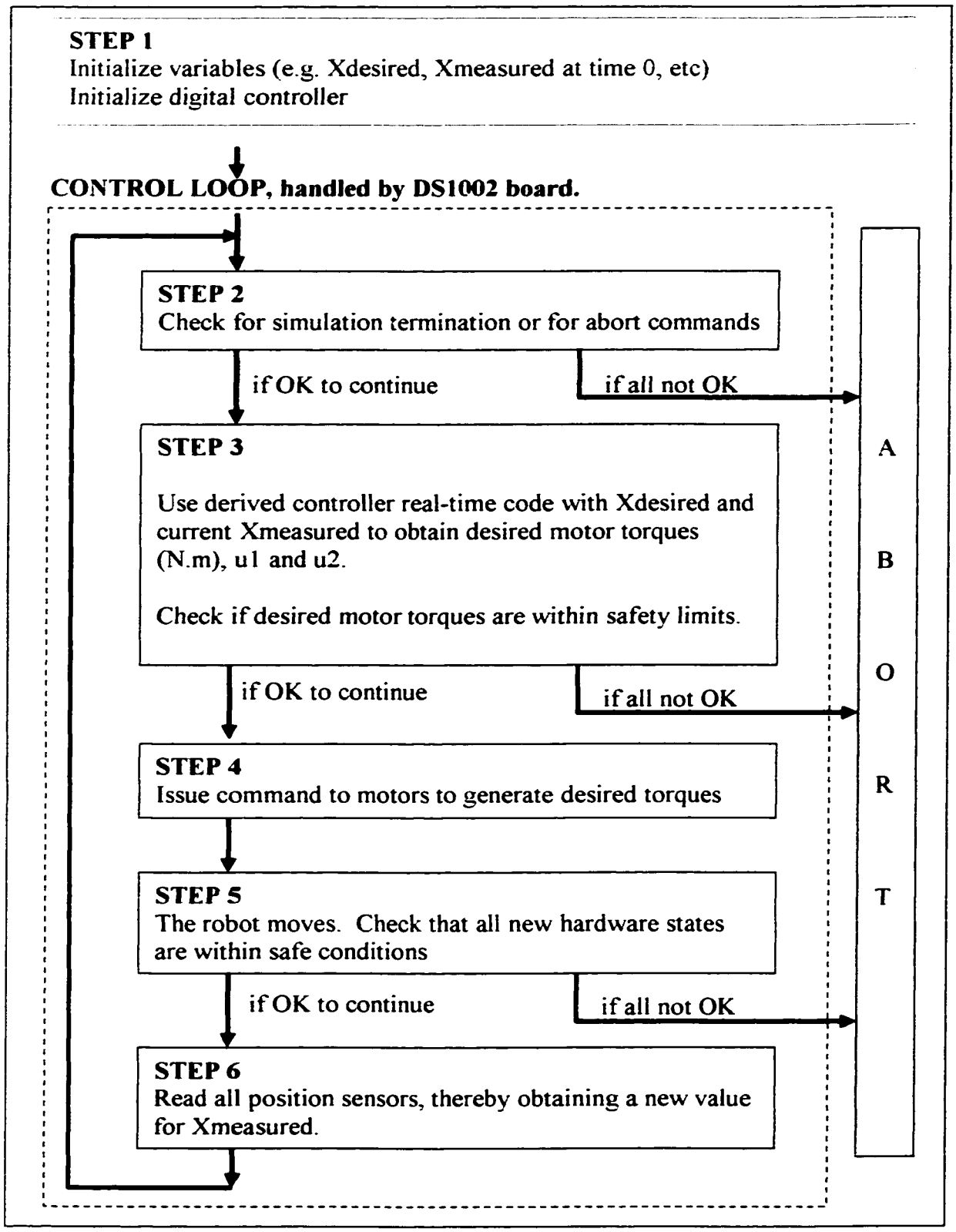
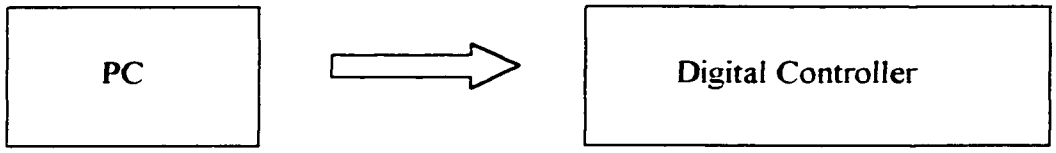


Figure 4-2: Control Algorithm

Step1: Initialize Variables; Initialize Digital Controller



Predefine the
final_time ' = final time to complete simulation
time_step ' = a value of 0.01 was used
steps_num ' = total number of steps until final_time

Predefine desired trajectory, by specifing the values of the angles of the links for each time step
q1_des(1..steps_num), q2_des(1..steps_num)
The desired motor angles are set to equal the desired link angles
qm1_des = q1_des, qm2_des = q2_des
The other four states (the motor and link velocities) are set to 0.
The eight states that make up the desired trajectory will be collectively referred to as Xdesired.

Initialize values for measured angles of links and motors to 0
qm1_mes = 0, qm2_mes = 0, q1_mes = 0, q2_mes = 0
The other four states are also initialized to zero, but and are never used by the controller. The eight states that make up the measured variables will be collectively referred to as Xmeasured.

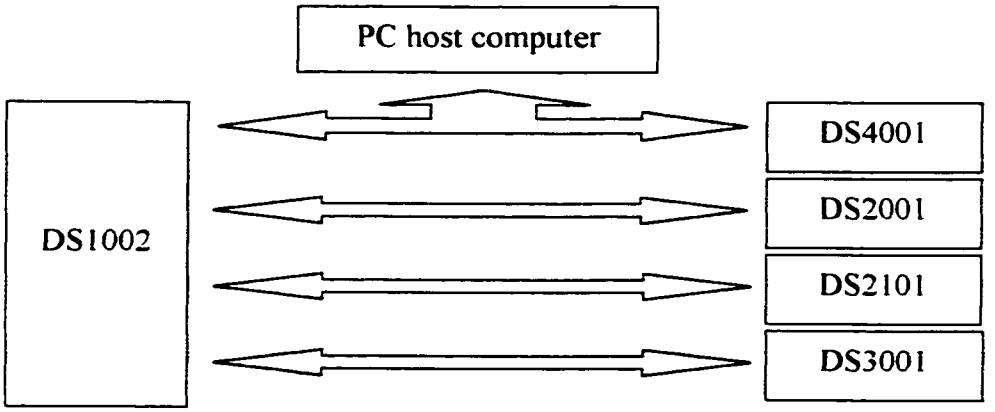
Initialize other necessary constants, for example
nm_to_volts1 ' constants of converting the desired motor torques (N.m) into
nm_to_volts2 ' actual voltage values to be sent to motors.

Send initialization commands to all digital controller.
Download C code into the digital controller.

Upon downloading the C code into the digital controller, these constants become stored in the memory of the DS1002 board.

Figure 4-3: Controller Algorithm Step 1 In More Detail

Step 2: Check boards' status, termination and abort conditions

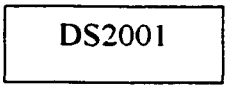


DS1002 board checks

- for termination of the downloaded simulation
- for abort commands issued by the PC host computer. Some of these commands are issued through the DS4001 board and other are issued directly to the DS1002 board.
- Status of all other digital boards

Figure 4-4: Controller Algorithm Step 2 In Detail

Step 3: Real-Time calculation of required motor torques for this instant



The real-time calculation of the required motor torques is carried out by the DS1002 card. The new motor torques will depend on the Xdesired and the current Xmeasured.

The DS1002 also checks that the required motor torques fall within the permissable range. The proposed controller was designed such that the required motor torques are always below the maximum possible values. If such a case exists, the DS1002 will abort the simulation and generate a warning.

Figure 4-5: Controller Algorithm Step 3 In Detail

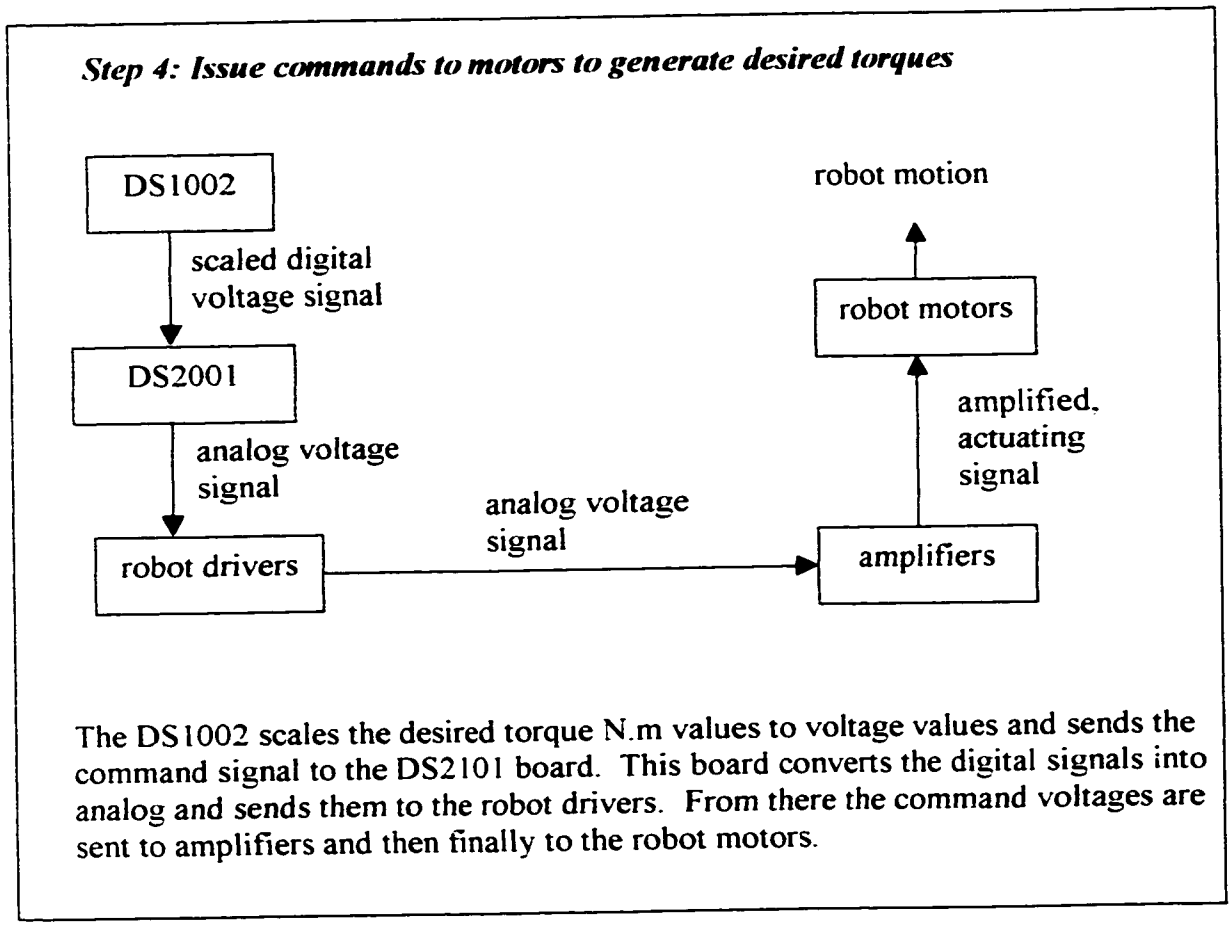
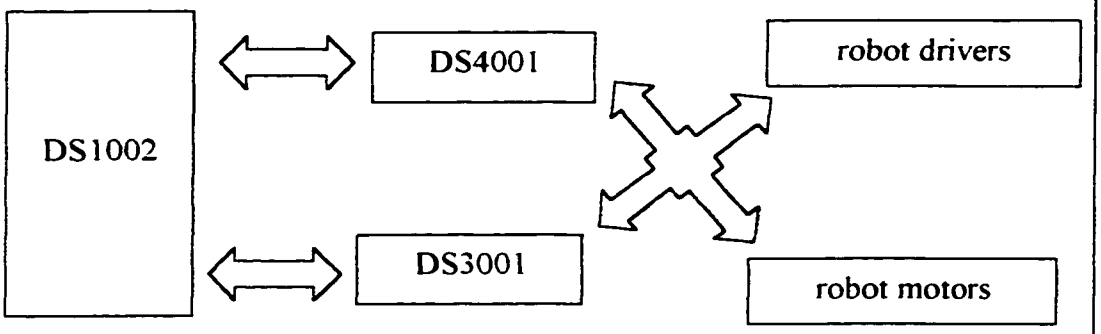


Figure 4-6: Controller Algorithm Step 4 In Detail

Step 5: Check that all new hardware states are within safe conditions



DS1002 board checks

- status of experimental hardware is checked (e.g. motor current overload ?, motion executed ? , driver overflow? driver overspeed ?, motion completed ?, etc.)
- the DS1002 performs the checks with data from the DS3001 and DS4001 cards, which in turn, receive data from the robot drivers and the robot motors

Figure 4-7: Control Algorithm Step 5 in Detail

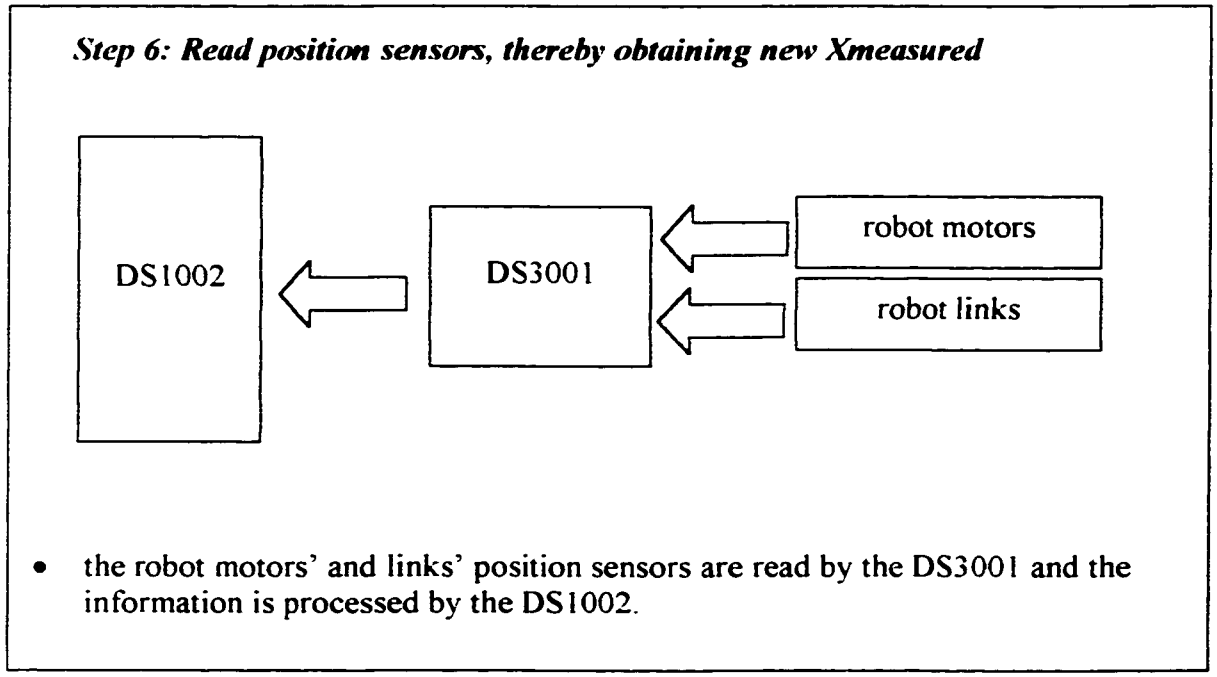


Figure 4-8: Control Algorithm Step 6 in Detail

4.4 CONCLUSIONS

This chapter describes the setup available for the experimental implementation of the proposed controller, and the proposed controller in algorithmic form. The research group at the IMS center will persue the implementation of these algorithms in the future.

Chapter 5: CONCLUSIONS

5.1 INTRODUCTION

Previous research has indicated that the rigid-robot dynamic models do not always hold true in industrial robot applications, and that the existent joint flexibility must be taken into account in order to achieve accurate robot position-tracking control.

Several flexible-joint robot dynamic models exist and several control schemes have been developed based on these dynamic models. Most of these existing controllers are nonlinear in nature and are complex to implement in industry. Ideally, controllers have the following properties: i) do not require full state feedback, ii) do not require noise-free measurements, and iii) are robust. However, existent nonlinear controllers for flexible-joint robots possess at most only two of these properties. Several linear controllers also exist, and others could be derived to possess all of the ideal-controller properties. However, because existing linear dynamic models are only accurate under limiting conditions, existing linear controllers are also limited in application.

This research started with the goals of deriving an acceptable and accurate linear dynamic model for flexible-joint robots and, based on this model, deriving a linear position-tracking controller for a study-case flexible-joint robots, which possess all of the above-mentioned controller properties.

5.2 ACHIEVEMENTS

In order to achieve the first goal of deriving an accurate linear dynamic model for flexible-joint robots, a more detailed nonlinear n-link dynamic model for flexible-joint robots was first proposed. This nonlinear model possessed additional terms which modeled friction by stick-slip functions and joint elasticity by nonlinear functions. A general n-link linear model was then proposed simply as the linearization of this proposed n-link nonlinear dynamic model. This linear model was already more detailed than existing linear dynamic models in that it was based on a more detailed nonlinear dynamic model and hence accounted in a more accurate way for more system dynamics. A version of this n-link linear model was then obtained for the study-case robot. During the linearization/optimization process, it turned out that the modeling of the friction by stick-slip function has one great advantage: it introduces several additional terms that can be varied during the optimization process. Making use of this advantage, a linear dynamic model was derived and proven to provide accurate tracking of the (true) nonlinear dynamic behavior over the entire range of operations.

This achievement permitted the derivation of a linear controller for the study-case robot on a solid foundation. Knowing that a controller robust towards plant disturbances could be built, the linear dynamic model was made more realistic by the derivation and inclusion of a stochastic plant-disturbance term. This plant disturbance term accounted for the difference between the linear dynamic model and the nonlinear dynamic model, and was obtained by empirical methods.

Based on this stochastic linear dynamic model, the controller design was then carried out. In order to further make the proposed controller more attractive to industry, only the position variables were chosen to be fed back, and it was assumed that these measurements were corrupted by noise. A robust path-tracking controller was then designed to work under these conditions through the LQG/LTR approach. The resulting controller was proven to provide good measurement-noise and plant-disturbance rejection, as well as accurate motion-tracking.

A third, originally unintended, contribution was made during the simulation process for the proposed linear model optimization and validation. The simulation in question was a time-response simulation carried out through a fourth-order Runge-Kutta integration routine. At that point it became apparent that the simulation routine for the nonlinear dynamic model would erroneously model the effects of static/coulomb or stick-slip friction, resulting in velocity time responses which had small, sharp oscillations about the 0 value, and which did not settle. This behavior is due the nature of the coulomb or stick-slip friction model which dictates a constant or increasing opposing friction force as the velocity approaches zero. Left uncorrected, this opposing friction force results in an acceleration of a magnitude large enough to change the sign of the velocity during the simulation time step. In this thesis, a corrective routine was proposed and derived, which allowed the Runge-Kutta simulation to correctly model the effects of static/coulomb or stick-slip friction.

5.3 DISCUSSION

The derivation of the proposed n-link linear dynamic model for flexible-joint robots is an important development in the study of robot dynamics, giving researchers two more parameters per robot joint that can be varied during the linearization/optimization phase. These two parameters are the derivatives of the stick-slip friction of the link and of the motor. It is expected that, with the aid of these additional parameters, acceptable and accurate linear models for other robot configurations can be derived. This allows for the application of a wider range of linear controllers for flexible-joint robots than was possible before.

The derivation of the proposed LQG/LTR controller serves as an example of this possibility and is an important achievement. It is an inexpensive controller, requiring only two position sensors per joint, and it is much easier to implement than most nonlinear controllers. It is also very robust and results in accurate motion tracking.

The derivation of the proposed Runge-Kutta simulation routine is important during the model development phase. Since stick-slip friction plays an important role in the model development, a corrective routine must always be included to model it properly. While other researchers may not use the Runge-Kutta integration routine, the flowcharts developed are helpful for any type of simulation.

5.4 FUTURE WORK

Perhaps the most important future work that should be carried out is the implementation of the developed controller on the study-case robot. Achieving this

laboratory implementation would give the work derived in this thesis much more scientific weight, and would bring it more to the attention of the industry.

In other aspects, as the literature survey of Chapter 1 indicates, the field of flexible-joint robots modeling and control is broad and the research presented in this thesis can be pursued in a variety of directions. It is the author's belief that this research could best be extended by the application of the models and control techniques derived in this thesis to different configurations of robots, specifically, for the case where the links are not planar. Such an application would better demonstrate the industrial relevance of this research. Another application the author believes demands more study is controller design for high speed applications. Much in today's industry is driven by production rates and, as a result, assembly robots are run at the highest speeds possible. Joint flexibility forces an upper bound on these high speeds as robots are forced to slow down to dampen out the resulting oscillations. Controlling these oscillations, through the control of the joint flexibility, would increase the upper bounds placed on the speeds.

Chapter 6: REFERENCES

1. Bedford A., and Fowler, W., "Dynamics: Engineering Mechanics", Addison-Wesley, New York, 1994, New York

2. Book, W.J., "Controlled Motion in an Elastic World", The Journal Of Dynamic Systems, Measurements And Control, Vol. 115, June 1993, pp. 252-261

3. Cesareo, G., and Marino, R., "On the Controllability Properties of Elastic Robots", 6th International Conference Analysis and Optimization of Systems, Nice, 1984.

4. Chen, G. R., and Malki, H. A., and Feigenspan, D., "Fuzzy PID Control of Temperature Stabilization and Flexible-Joint Robot Arm with Time-Varying Loads", College of Technology, UH, http://129.7.21.70/enl/news33/en33p3.html, August 98

5. De Luca, A., "Dynamic Control of Robots with Joint Elasticity", Proceedings of IEEE Conference on Robotics and Automation, Philadelphia, PA, April 1988, pp. 98-135

6. De Simone, C., and Nicolo, F., "On the Control of Elastic Robots by Feedback Decoupling", IASTED International Journal of Robotics and Automation, 1,2, 1986

7. De Wit, C., and Noël, P., and Aubin, A., and Brogliato, B., "Adaptive Friction Compensation in Robot Manipulators: Low Velocities", The International Journal Of Robotics Research, Vol. 10, No. 3, June 1991, pp. 189-199

8. dSPACE digital signal processing and control engineering GmbH, DSP-CITpro Hardware Version 1.0 and related documentation, Paderborn, Germany, 1990

9. Dupont, P. E., "The Effect of Friction on the Forward Dynamics Problem", The International Journal Of Robotics Research, Vol. 12, No. 2, April 1993, pp. 164-179

10. ElMaraghy, W.H., and ElMaraghy, H. A., and Zaki, A., and Massoud, A., "A Study On The Design And Control Of Robot Manipulators With Flexibilities", Symposium on Robot Control, September 19, 1994, Capri, Italy, pp. 495-501

11. Ghorbel, F., and Spong, M.W., "Stability Analysis of Adaptively Controlled Flexible Joint Manipulators", Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, 1990, pp. 2538-2544

12. Good, M.C., and Sweet, L. M., and Strobel, K. L., "Dynamic Models for Control Systems Design of Integrated Robot and Drive Systems", The Journal Of Dynamic Systems, Measurements And Control, Vol. 107, March 1985, pp. 53-59

13. Haessig, Jr., D . A., and Friedland, B., "On the Modeling and Simulation of Friction", The Journal Of Dynamic Systems, Measurements And Control, Vol. 113, September 1991, pp. 354-362

14. Hsiao, M., and Huang, J., and Cox, D. E., "Iterative LQG Controller Design Through Closed-Loop Identification", Journal of Dynamic Systems, Measurement, and Control, Vol. 118, June 1996, pp. 366-372

15. Jankowski, K. P., and ElMaraghy, H. A., and ElMaraghy, "Inverse Dynamics Control of Multiple Robot Arms with Flexible Joints", Proceedings IEEE Conference on Robotics and Automation, Atlanta, GA, May, 1993, pp. 996-1003

16. Jazwinski, A. H., "Stochastic Process and Filtering Theory", Academic Press, NY, 1970

17. Lahdhiri, T., and ElMaraghy, H. A. , "Optimal Nonlinear Position Tracking Control of a Two-Link Flexible-Joint Robot Manipulator", Proceedings of the Fifth International Symposium on Experimental Robotics, Barcelona, Spain, June 1997, pp. 407-418

18. Lammerts, I.M.M, and Veldpaus, F.E., and Vand de Molengraft, M.J.G., and Kok, J.J., "Adaptive Computer Reference Computed Torque Control of Flexible Robots", Journal of Dynamic Systems, Measurement, and Control, Vol. 117, March 1995, pp. 31-36

19. Lin Y. and Yu, A., "Linear Robust Trajectory Control of Flexible Joint Manipulators", Robotica, Vol. 14, 1996, pp. 34-41

20. Maciejowski, J. M., "Multivariable Feedback Design", Addison-Wesley, Ontario, 1989

21. Massoud, A., "Motion and Force Control of Flexible Joint Robot Manipulators", Ph.D. Dissertation, University of McMaster, 1994

22. MathWorks, Inc., "Matlab v5.1", U.S.A. , 1997

23. Nicosia, S., and Tomei, P., "A Tracking Controller for Flexible Joint Robots Using Only Link Position Feedback", IEEE Transactions on Automatic Control, May 1995, pp. 885-890

24. Nicosia, S. and Tomei, P., and Tornambe A., "A Nonlinear Observer for Elastic Robots", IEEE Journal of Robotics and Automation, Vol. 4, No. 1, February 1988, pp. 45-51

25. Qu, Z., "Input-Output Robust Tracking Control Design for Flexible Joint Robots", IEEE Transactions on Automatic Control, Vol. 40, No. 1, January 1995, pp. 78-83

26. Sciavicco, L., and Siciliano, B., "Modeling and Control of Robot Manipulators", McGraw-Hill Companies, Inc., 1996

27. Slotine J.-J.E. and Hong, S., "Two-Time Scale Sliding Control of Manipulators with Flexible Joints", American Control Conference, Seattle, 1986, pp. 805-810

28. Spong, M.W., "On the Force Control Problem for Flexible Joint Manipulators", IEEE Transactions on Automatic Control, Vol. 34, No. 1, January 1989, pp. 107-111

29. Spong, M. W., "Modeling and Control of Elastic Joint Robots", The Journal Of Dynamic Systems, Measurements And Control, Vol. 109, December 1987, pp. 310-319

30. Sweet, L. M., and Good, M. C., "Redefinition of the Robot Motion-Control Problem", IEEE Control Systems Magazine, August 1985, pp. 18-25

31. Tomei, P., "A Simple PD Controller for Robots with Elastic Joints", IEEE Transactions on Automatic Control, Vol. 36, No. 10, October 1991, pp. 1208-1213

32. Tomei, P., "An Observer for Flexible Joint Robots", IEEE Transactions on Automatic Control, Vol. 35, No. 6, June 1990, pp. 739-743

33. Tomei, P., "Tracking Control of Flexible Joint Robots with Uncertain Parameters and Disturbances", IEEE Transactions on Automatic Control, Vol. 39, No. 5, May 1994, pp. 1037-1072

34. Whittaker, W. L., and Kanade, T., and Allen, P., and Bejczy, A. K., and Lowrie, J. W., and McCain, H. G., and Montemerlo, M. D., and Sheridan, T. B., "JTEC Panel Report on Space Robotics in Japan", Japanese Technology Evaluation Center, 1991

35. Yueh-Jaw Lin, and Aiping Yu, "Linear Robust Trajectory Control of Flexible Joint Manipulators", Robotica, Vol. 14, 1996, pp. 375-379

APPENDIX A: STUDY-CASE ROBOT MODEL

A.1 INTRODUCTION

This section contains the details of the flexible-joint robot, introduced in Chapter 1, and used as the study case for the research presented in this thesis. Section A.2 contains the robot physical data. Section A.3 contains robot-specific functions, namely the nonlinear joint-elasticity functions $K_F(.)$ and the stick-slip friction functions, $F_{CL}(.)$ and $F_{CM}(.)$, appearing in Chapter 2. Finally, section A.4 contains the linear state-space dynamic model for this study case robot, derived from the equations presented in Chapter 2.

A.2 EXPERIMENTAL ROBOT PARAMETERS

The parameters of this robot system, presented in Table (A-1), were obtained from several sources: original construction specifications, through a sine sweep performed on this system by Massoud '94 [21], and through I-DEAS™ modeling.

parameter	symbol	value
distance between origin of link 1 and its cg.	a_1	0.281 m
distance between origin of link 2 and its c. of g.	a_2	0.113 m
length of robot link 1	L_1	0.40 m
length of robot link 2	L_2	0.35 m
moment of inertia of link 1	I_1	1.4714
moment of inertia of link 2	I_2	0.0793
mass of link 1	m_1	10.38 kg
mass of link 2	m_2	2.324 kg
mass of rotor 1	mm_1	8.4 kg
mass of rotor 2	mm_2	3.4 kg
moment of inertia of rotor 1	I_{M1}	0.1224 kg.m^2
moment of inertia of rotor 2	I_{M2}	0.0168 kg.m^2
viscous friction, motor 1	b_{M1}	1.254 N.m.s/rad
viscous friction, motor 2	b_{M2}	0.119 N.m.s/rad
viscous friction, link 1	b_{L1}	2.041 N.m.s/rad
viscous friction, link 2	b_{L2}	0.242 N.m.s/rad

Table A-1: Experimental Robot Parameters

A.3 ROBOT-SPECIFIC FUNCTIONS

Ideally, expressions for the robot-specific functions $\mathbf{K_F(.)}$, $\mathbf{F_{CL}(.)}$, and $\mathbf{F_{CM}(.)}$, of the proposed nonlinear dynamic model (2-5)-(2-6), should be determined experimentally. However, as indicated in Chapter 4, for the work described in this thesis, the actual robot could not be operated due to the lack of certain software modules. Without experimental data available, the pre-existing data was used instead to derive these robot-specific functions.

The stick-slip friction models were derived using previously recorded, experimentally observed static and coulomb values for this robot, by Massoud '94 [21]. Attempts were made to employ Equation (2-3) to achieve stick-slip friction functions that closely approximate the available static and coulomb friction values. However, a combination of the curve parameters φ of Equation (2-3), that leads to a satisfactory approximation with a relatively steep change from the static friction to the coulomb friction mode, was not found. The final model, presented in Figure (A-1), was achieved using the following criteria:

$$\text{if } |\dot{q}| \leq 1°/\text{sec}, \text{ stick slip friction} = F_{\text{stick-slip}}(\dot{q}) \tag{A-1}$$

$$\text{if } |\dot{q}| > 1°/\text{sec}, \text{ stick slip friction} = F_{\text{stick-slip}}(\text{sign}(\dot{q}) \times 1°/\text{sec}) \tag{A-2}$$

where $F_{\text{stick-slip}}$ is given by Equation (2-3) and its parameters φ were chosen such that

$$F_{\text{stick-slip}}(\text{sign}(\dot{q}) \times 1°/\text{sec}) \approx \text{the experimental coulomb friction of [2]} \tag{A-3}$$

All stick-slip friction functions were assumed to have the same shape and only differ in amplitude, which was achieved by the introduction of a scaling factor.

Cubic functions were selected to model the joint elasticity. These functions are the most widely used nonlinear functions to model the deflection-torque curves of torsional springs (Bedford and Fowler, '94 [1]). The cubic functions selected to model the flexible joints are shown in Figure (A-2).

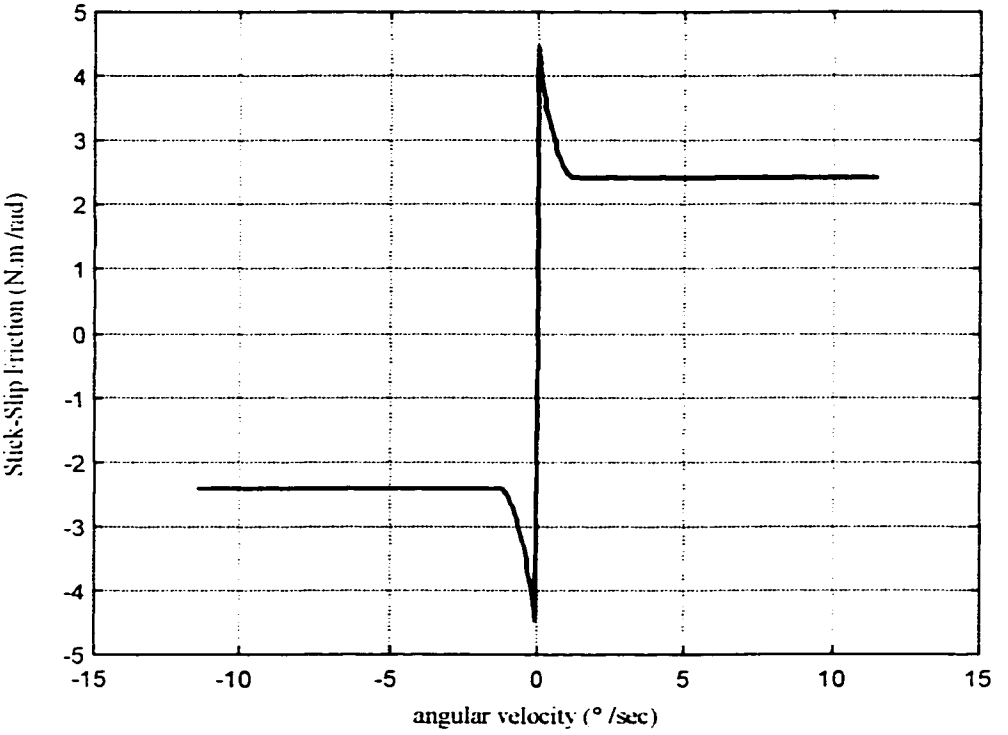
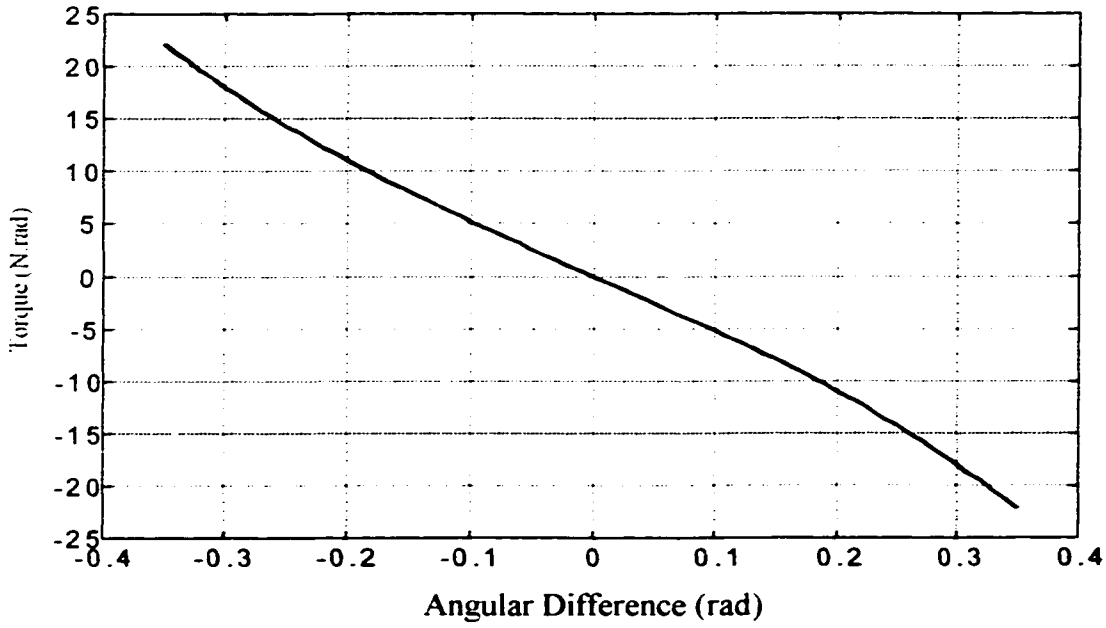


Figure A-1: Proposed Stick-Slip Friction Model

(a) Joint 1: k_{t1}



(b) Joint 2: k_{t2}

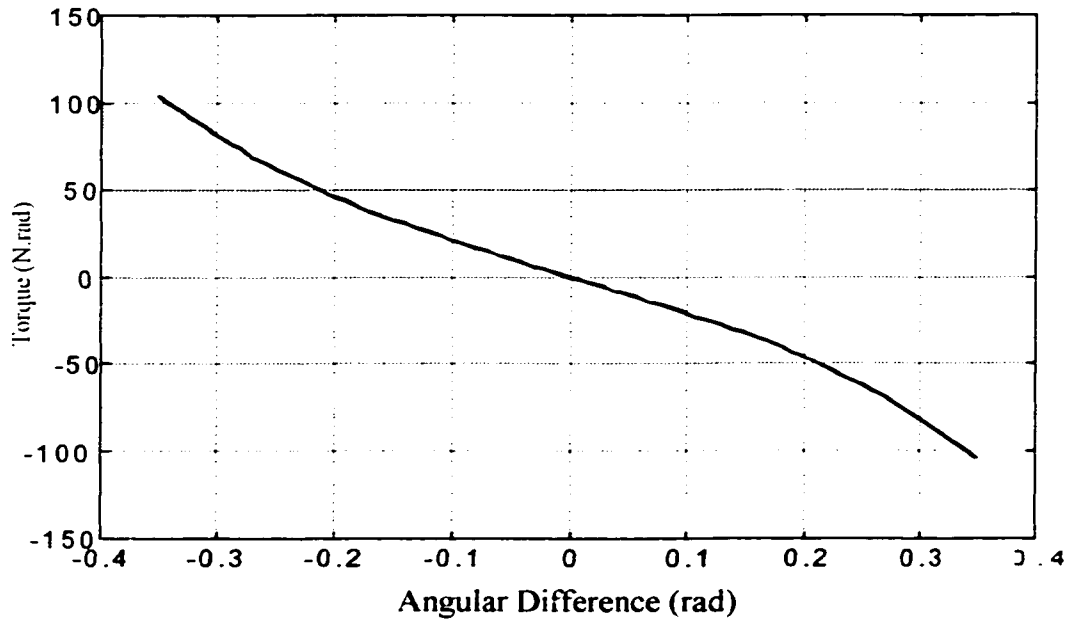


Figure A-2: Proposed Nonlinear Joint Elasticity Curves

A.4 EXPLICIT EQUATIONS FOR THE STUDY-CASE LINEAR DYNAMIC MODEL

Based on the robot parameters given in Table (A-1), the terms $\mathbf{I_M}$, $\mathbf{\Lambda}$, $\mathbf{D(.)}$, $\mathbf{C(.)}$, $\mathbf{F_{CM}(.)}$, $\mathbf{F_{CM}(.)}$, $\mathbf{K_F}$, of the nonlinear dynamic model (2-4)-(2-5) are given by:

$$\mathbf{I_M} = \begin{bmatrix} I_{m1} & 0 \\ 0 & I_{m2} \end{bmatrix}, \ \mathbf{\Lambda} = \begin{bmatrix} 0 & I_{m2} \\ 0 & 0 \end{bmatrix}, \ \mathbf{B_L} = \begin{bmatrix} \mathbf{b}_{L1} & \\ & \mathbf{b}_{L2} \end{bmatrix}, \ \mathbf{B_M} = \begin{bmatrix} \mathbf{b}_{M1} & \\ & \mathbf{b}_{M2} \end{bmatrix} \tag{A-4}$$

$$\mathbf{F_{CM}(x_2)} = \begin{bmatrix} f_{CM1}(\dot{q}_{M1}) \\ f_{CM2}(\dot{q}_{M2}) \end{bmatrix}, \ \mathbf{F_{CL}(x_4)} = \begin{bmatrix} f_{CL1}(\dot{q}_1) \\ f_{CL2}(\dot{q}_2) \end{bmatrix}, \ \mathbf{K_f(x_1, x_3)} = \begin{bmatrix} k_{f1}(q_{M1}, q_1) \\ k_{f2}(q_{M2}, q_2) \end{bmatrix} \tag{A-5}$$

$$\mathbf{D(q)} = \left[\begin{array}{c|c} d_1 + 2 \times d_2 \times \cos(q_2) & d_3 + d_2 \times \cos(q_2) \\ \hline d_3 + d_2 \times \cos(q_2) & d_3 \end{array}\right] \tag{A-6}$$

$$\mathbf{C(q, \dot{q})} = \begin{bmatrix} -2 \times d_2 \times \sin(q_2) \times \dot{q}_2 \times \dot{q}_1 - d_2 \times \sin(q_2) \times (\dot{q}_2)^2 \\ d_2 \times (\dot{q}_1)^2 \times \sin(q_2) \end{bmatrix} \tag{A-7}$$

where the functions $f_{CM1}(.)$, $f_{CM2}(.)$, $f_{CL1}(.)$, $f_{CL2}(.)$ of Equation (A-5) have the same shape given in Figure (A-1), the functions $k_{f1}(.)$ and $k_{f2}(.)$ of Equation (A-5) are shown in Figure (A-2), and d_1, d_2, and d_3 of Equations (A-6) and (A-7) are given by :

$$d_1 = I_1 + I_2 + m_1 \times (a_1)^2 + mm_2 \times (L_1)^2 + m_2 \times [(L_1)^2 + (a_2)^2] \tag{A-8}$$

$$d_2 = m_2 \times L_1 \times a_2, \ \ d_3 = I_2 + m_2 \times (a_2)^2 \tag{A-9}$$

The states of this system are $\mathbf{x}_1 = [q_{M1} q_{M2}]^T$, $\mathbf{x}_2 = [\dot{q}_{M1} \dot{q}_{M2}]^T$, $\mathbf{x}_3 = [q_1 \ q_2]^T$, $\mathbf{x}_4 = [\dot{q}_1 \ \dot{q}_2]^T$ and the input $\mathbf{U}$ = diag(u_1, u_2), where u_1 and u_2 are the torques applied by the motors to joint 1 and 2, respectively. In this case, the matrices **A** and **B** of Equation (2-17) become

$$\mathbf{A} = \left[\begin{array}{cccc} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_4} \\ \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_4} \\ \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_2} & \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_3} & \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_4} \\ \frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_2} & \frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_3} & \frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_4} \end{array}\right]_{\text{at (X0,U0)}}, \quad \mathbf{B} = \left[\begin{array}{c} \frac{\partial \mathbf{f}_1}{\partial \mathbf{u}} \\ \frac{\partial \mathbf{f}_2}{\partial \mathbf{u}} \\ \frac{\partial \mathbf{f}_3}{\partial \mathbf{u}} \\ \frac{\partial \mathbf{f}_4}{\partial \mathbf{u}} \end{array}\right]_{\text{at (X0,U0)}} \tag{A-10}$$

The partial derivative terms $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1}$, $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2}$, $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3}$, and $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_4}$ of Equation (A-10) are given by

$$\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} = \left[\frac{\partial \mathbf{f}_1}{\partial \mathbf{q}_{M1}} \quad \frac{\partial \mathbf{f}_1}{\partial \mathbf{q}_{M2}}\right] = [\mathbf{0}]_{2\times 2} \qquad \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2} = \left[\frac{\partial \mathbf{f}_1}{\partial \dot{\mathbf{q}}_{M1}} \quad \frac{\partial \mathbf{f}_1}{\partial \dot{\mathbf{q}}_{M2}}\right] = [\mathbf{I}]_{2\times 2}, \tag{A-11}$$

$$\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3} = \left[\frac{\partial \mathbf{f}_1}{\partial \mathbf{q}_1} \quad \frac{\partial \mathbf{f}_1}{\partial \mathbf{q}_2}\right] = [\mathbf{0}]_{2\times 2}, \qquad \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_4} = \left[\frac{\partial \mathbf{f}_1}{\partial \dot{\mathbf{q}}_1} \quad \frac{\partial \mathbf{f}_1}{\partial \dot{\mathbf{q}}_2}\right] = [\mathbf{0}]_{2\times 2}, \tag{A-12}$$

The partial derivatives term $\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1}$ of Equation (A-10) is given by

$$\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} = \left[\frac{\partial \mathbf{f}_2}{\partial \mathbf{q}_{M1}} \quad \frac{\partial \mathbf{f}_2}{\partial \mathbf{q}_{M2}}\right]_{2\times 2}, \tag{A-13}$$

where the partial derivatives $\frac{\partial \mathbf{f}_2}{\partial \mathbf{q}_{M1}}$ and $\frac{\partial \mathbf{f}_2}{\partial \mathbf{q}_{M2}}$ are given by:

$$\frac{\partial \mathbf{f}_2}{\partial \mathbf{q}_{M1}} = \left[\alpha_1 \times \left(\frac{\partial \mathbf{k}_f}{\partial \mathbf{q}_{M1}} + \mathbf{\Lambda}^T \times \mathbf{D}^{-1} \times \frac{\partial \mathbf{k}_f}{\partial \mathbf{q}_{M1}}\right)\right]_{\text{at (X0,U0)}} \tag{A-14}$$

$$\frac{\partial \mathbf{f}_2}{\partial \mathbf{q}_{M2}} = \left[\alpha_1 \times \left(\frac{\partial \mathbf{k}_f}{\partial \mathbf{q}_{M2}} + \mathbf{\Lambda}^T \times \mathbf{D}^{-1} \times \frac{\partial \mathbf{k}_f}{\partial \mathbf{q}_{M2}}\right)\right]_{\text{at (X0,U0)}} \tag{A-15}$$

The partial derivatives term $\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2}$ of Equation (A-10) is given by

$$\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2} = \left[\frac{\partial \mathbf{f}_2}{\partial \dot{\mathbf{q}}_{M1}} \quad \frac{\partial \mathbf{f}_2}{\partial \dot{\mathbf{q}}_{M2}} \right]_{2 \times 2} , \tag{A-16}$$

where the partial derivatives $\frac{\partial \mathbf{f}_2}{\partial \dot{\mathbf{q}}_{M1}}$ and $\frac{\partial \mathbf{f}_2}{\partial \dot{\mathbf{q}}_{M2}}$ are given by:

$$\frac{\partial \mathbf{f}_2}{\partial \dot{\mathbf{q}}_{M1}} = \left[\boldsymbol{\alpha}_1 \times \left(-\frac{\partial \mathbf{F}_{CM}}{\partial \dot{\mathbf{q}}_{M1}} - \mathbf{B}_M(.,1) \right) \right]_{at(X0,U0)} , \tag{A-17}$$

$$\frac{\partial \mathbf{f}_2}{\partial \dot{\mathbf{q}}_{M2}} = \left[\boldsymbol{\alpha}_1 \times \left(-\frac{\partial \mathbf{F}_{CM}}{\partial \dot{\mathbf{q}}_{M2}} - \mathbf{B}_M(.,2) \right) \right]_{at(X0,U0)} \tag{A-18}$$

The partial derivatives term $\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_3}$ of Equation (A-10) is given by

$$\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_3} = \left[\frac{\partial \mathbf{f}_2}{\partial q_1} \quad \frac{\partial \mathbf{f}_2}{\partial q_2} \right]_{2 \times 2} , \tag{A-19}$$

where the partial derivatives $\frac{\partial \mathbf{f}_2}{\partial q_1}$ and $\frac{\partial \mathbf{f}_2}{\partial q_2}$ are given by:

$$\frac{\partial \mathbf{f}_2}{\partial q_1} = \left[\begin{array}{l} \frac{\partial \boldsymbol{\alpha}_1}{\partial q_1} \times \left(-\mathbf{F}_{CM} + \mathbf{k}_f - \mathbf{B}_M \times \mathbf{q}_M + \boldsymbol{\alpha}_3 \times \left(\mathbf{B}_L \times \mathbf{q} + \mathbf{C} + \mathbf{k}_f + \mathbf{F}_{CL} \right) \right) + \\ + \boldsymbol{\alpha}_1 \times \left(\begin{array}{l} \left(\frac{\partial \mathbf{k}_f}{\partial q_1} + \frac{\partial \boldsymbol{\alpha}_3}{\partial q_1} \times (\mathbf{B}_L \times \mathbf{q} + \mathbf{C} + \mathbf{k}_f + \mathbf{F}_{CL}) + \right. \\ \left. + \boldsymbol{\alpha}_3 \times \left(\mathbf{B}_L(.,1) + \frac{\partial \mathbf{C}}{\partial q_1} + \frac{\partial \mathbf{k}_f}{\partial q_1} \right) \right) + \frac{\partial \boldsymbol{\alpha}_1}{\partial q_1} \times \mathbf{U} \end{array} \right]_{at(X0,U0)} \end{array} \right. \tag{A-20}$$

$$\frac{\partial \mathbf{f}_2}{\partial q_2} = \left[\begin{array}{l} \frac{\partial \boldsymbol{\alpha}_1}{\partial q_2} \times \left(-\mathbf{F}_{CM} + \mathbf{k}_f - \mathbf{B}_M \times \mathbf{q}_M + \boldsymbol{\alpha}_3 \times \left(\mathbf{B}_L \times \mathbf{q} + \mathbf{C} + \mathbf{k}_f + \mathbf{F}_{CL} \right) \right) + \\ + \boldsymbol{\alpha}_1 \times \left(\begin{array}{l} \left(\frac{\partial \mathbf{k}_f}{\partial q_2} + \frac{\partial \boldsymbol{\alpha}_3}{\partial q_2} \times (\mathbf{B}_L \times \mathbf{q} + \mathbf{C} + \mathbf{k}_f + \mathbf{F}_{CL}) + \right. \\ \left. + \boldsymbol{\alpha}_3 \times \left(\mathbf{B}_L(.,2) + \frac{\partial \mathbf{C}}{\partial q_2} + \frac{\partial \mathbf{k}_f}{\partial q_2} \right) \right) + \frac{\partial \boldsymbol{\alpha}_1}{\partial q_2} \times \mathbf{U} \end{array} \right]_{at(X0,U0)} \end{array} \right. \tag{A-21}$$

In Equation (A-20)-(A-21), $\alpha_3(\mathbf{q})$ is a nonlinear matrix of the size of **D(.)** and is defined by

$$\alpha_3(\mathbf{q}) = \Lambda^T \times [\mathbf{D}(\mathbf{q})]^{-1} \tag{A-22}$$

The partial derivatives term $\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_4}$ of Equation (A-10) is given by

$$\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_4} = \left[\frac{\partial \mathbf{f}_2}{\partial \dot{q}_1} \quad \frac{\partial \mathbf{f}_2}{\partial \dot{q}_2} \right]_{2 \times 2}, \tag{A-23}$$

where the partial derivatives $\frac{\partial \mathbf{f}_2}{\partial \dot{q}_1}$ and $\frac{\partial \mathbf{f}_2}{\partial \dot{q}_2}$ are given by

$$\frac{\partial \mathbf{f}_2}{\partial \dot{q}_1} = \left[\alpha_1 \times \alpha_3 \times \left(\frac{\partial \mathbf{F}_{CL}}{\partial \dot{q}_1} + \mathbf{B}_L(.,1) + \frac{\partial \mathbf{C}}{\partial \dot{q}_1} \right) \right]_{at(X0,U0)} \tag{A-24}$$

$$\frac{\partial \mathbf{f}_2}{\partial \dot{q}_2} = \left[\alpha_1 \times \alpha_3 \times \left(\frac{\partial \mathbf{F}_{CL}}{\partial \dot{q}_2} + \mathbf{B}_L(.,2) + \frac{\partial \mathbf{C}}{\partial \dot{q}_2} \right) \right]_{at(X0,U0)} \tag{A-25}$$

The partial derivatives terms $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1}$, $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2}$, $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3}$, and $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_4}$ of Equation (A-10) are given by

$$\frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_1} = \left[\frac{\partial \mathbf{f}_3}{\partial q_{M1}} \quad \frac{\partial \mathbf{f}_3}{\partial q_{M2}} \right]_{k \times k} = [0]_{k \times k}, \qquad \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_2} = \left[\frac{\partial \mathbf{f}_3}{\partial \dot{q}_{M1}} \quad \frac{\partial \mathbf{f}_3}{\partial \dot{q}_{M2}} \right]_{2 \times 2} = [\mathbf{I}]_{2 \times 2} \tag{A-26}$$

$$\frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_3} = \left[\frac{\partial \mathbf{f}_3}{\partial q_1} \quad \frac{\partial \mathbf{f}_3}{\partial q_2} \right]_{2 \times 2} = [0]_{2 \times 2}, \qquad \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_4} = \left[\frac{\partial \mathbf{f}_3}{\partial \dot{q}_1} \quad \frac{\partial \mathbf{f}_3}{\partial \dot{q}_2} \right]_{2 \times 2} = [0]_{2 \times 2} \tag{A-27}$$

The partial derivatives term $\frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_1}$ of Equation (A-10) is given by

$$\frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_1} = \left[\frac{\partial \mathbf{f}_4}{\partial q_{M1}} \quad \frac{\partial \mathbf{f}_4}{\partial q_{M2}} \right]_{2 \times 2}, \tag{A-28}$$

where the partial derivatives $\frac{\partial \mathbf{f}_4}{\partial \mathbf{q}_{M1}}$ and $\frac{\partial \mathbf{f}_4}{\partial \mathbf{q}_{M2}}$ are given by

$$\frac{\partial \mathbf{f}_4}{\partial \mathbf{q}_{M1}} = \left[\boldsymbol{\alpha}_2 \times \left(\frac{\partial \mathbf{k}_r}{\partial \mathbf{q}_{M1}} + \boldsymbol{\Lambda}^T \times [\mathbf{I}_M]^{-1} \times \frac{\partial \mathbf{k}_r}{\partial \mathbf{q}_{M1}} \right) \right]_{at\,(X0,U0)} \tag{A-29}$$

$$\frac{\partial \mathbf{f}_4}{\partial \mathbf{q}_{M2}} = \left[\boldsymbol{\alpha}_2 \times \left(\frac{\partial \mathbf{k}_r}{\partial \mathbf{q}_{M2}} + \boldsymbol{\Lambda}^T \times [\mathbf{I}_M]^{-1} \times \frac{\partial \mathbf{k}_r}{\partial \mathbf{q}_{M2}} \right) \right]_{at\,(X0,U0)} \tag{A-30}$$

The partial derivatives term $\frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_2}$ of Equation (A-10) is given by

$$\frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_2} = \left[\frac{\partial \mathbf{f}_4}{\partial \dot{\mathbf{q}}_{M1}} \quad \frac{\partial \mathbf{f}_4}{\partial \dot{\mathbf{q}}_{M2}} \right]_{2 \times 1}, \tag{A-31}$$

where the partial derivatives $\frac{\partial \mathbf{f}_4}{\partial \dot{\mathbf{q}}_{M1}}$ and $\frac{\partial \mathbf{f}_4}{\partial \dot{\mathbf{q}}_{M2}}$ are given by

$$\frac{\partial \mathbf{f}_4}{\partial \dot{\mathbf{q}}_{M1}} = \left[\boldsymbol{\alpha}_2 \times \boldsymbol{\Lambda} \times [\mathbf{I}_M]^{-1} \times \left(-\frac{\partial \mathbf{F}_{CM}}{\partial \dot{\mathbf{q}}_{M1}} - \mathbf{B}_M(.,1) \right) \right]_{at\,(X0,U0)} \tag{A-32}$$

$$\frac{\partial \mathbf{f}_4}{\partial \dot{\mathbf{q}}_{M2}} = \left[\boldsymbol{\alpha}_2 \times \boldsymbol{\Lambda} \times [\mathbf{I}_M]^{-1} \times \left(-\frac{\partial \mathbf{F}_{CM}}{\partial \dot{\mathbf{q}}_{M2}} - \mathbf{B}_M(.,2) \right) \right]_{at\,(X0,U0)} \tag{A-33}$$

The partial derivatives term $\frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_3}$ of Equation (A-10) is given by

$$\frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_3} = \left[\frac{\partial \mathbf{f}_4}{\partial \mathbf{q}_1} \quad \frac{\partial \mathbf{f}_4}{\partial \mathbf{q}_2} \right]_{2 \times 2}, \tag{A-34}$$

where the partial derivatives $\frac{\partial \mathbf{f}_4}{\partial \mathbf{q}_1}$ and $\frac{\partial \mathbf{f}_4}{\partial \mathbf{q}_2}$ are given by

$$\frac{\partial \mathbf{f}_4}{\partial q_1} = \left[\begin{array}{l} \dfrac{\partial \alpha_2}{\partial q1}\left(\mathbf{F}_{CL} + \mathbf{k}_f + \mathbf{C} + \mathbf{B}_L \times \dot{\mathbf{q}} + \Lambda \times [\mathbf{I}_M]^{-1} \times \left(\mathbf{k}_f - \mathbf{F}_{CM} - \mathbf{B}_M \times \dot{\mathbf{q}}_M\right)\right) + \\ +\alpha 2 \times \left(\dfrac{\partial \mathbf{F}_{CL}}{\partial q_1} + \dfrac{\partial \mathbf{k}_f}{\partial q_1} + \dfrac{\partial \mathbf{C}}{\partial q_1} \times \Lambda \times [\mathbf{I}_M]^{-1} \times \dfrac{\partial \mathbf{k}_f}{\partial q_1} \right) + \dfrac{\partial \alpha_2}{\partial q_1} \times \Lambda \times [\mathbf{I}_M]^{-1} \times \mathbf{U} \end{array} \right]_{at(X0,U0)} \quad \text{(A-35)}$$

$$\frac{\partial \mathbf{f}_4}{\partial q_2} = \left[\begin{array}{l} \dfrac{\partial \alpha_2}{\partial q_2} \times \left(\mathbf{F}_{CL} + \mathbf{k}_f + \mathbf{C} + \mathbf{B}_L \times \dot{\mathbf{q}} + \Lambda \times [\mathbf{I}_M]^{-1} \times \left(\mathbf{k}_f - \mathbf{F}_{CM} - \mathbf{B}_M \times \dot{\mathbf{q}}_M\right)\right) + \\ +\alpha 2 \times \left(\dfrac{\partial \mathbf{F}_{CL}}{\partial q_2} + \dfrac{\partial \mathbf{k}_f}{\partial q_2} + \dfrac{\partial \mathbf{C}}{\partial q_2} \times \Lambda \times [\mathbf{I}_M]^{-1} \times \dfrac{\partial \mathbf{k}_f}{\partial q_2} \right) + \dfrac{\partial \alpha_2}{\partial q_2} \times \Lambda \times [\mathbf{I}_M]^{-1} \times \mathbf{U} \end{array} \right]_{at(X0,U0)} \quad \text{(A-36)}$$

The partial derivatives term $\dfrac{\partial \mathbf{f}_4}{\partial \mathbf{x}_4}$ of Equation (A-10) is given by

$$\frac{\partial \mathbf{f}_4}{\partial \mathbf{x}_4} = \left[\frac{\partial \mathbf{f}_4}{\partial \dot{q}_1} \quad \frac{\partial \mathbf{f}_4}{\partial \dot{q}_2} \right]_{2 \times 2}, \quad \text{(A-37)}$$

where the partial derivatives $\dfrac{\partial \mathbf{f}_4}{\partial \dot{q}_1}$ and $\dfrac{\partial \mathbf{f}_4}{\partial \dot{q}_2}$ of Equation (A-10) are given by

$$\frac{\partial \mathbf{f}_4}{\partial \dot{q}_1} = \left[\alpha_2 \times \left(\frac{\partial \mathbf{F}_{CL}}{\partial \dot{q}_1} + \mathbf{B}_L(.,1) + \frac{\partial \mathbf{C}}{\partial \dot{q}_1} \right) \right]_{2 \times 1}, \quad \text{(A-38)}$$

$$\frac{\partial \mathbf{f}_4}{\partial \dot{q}_2} = \left[\alpha_2 \times \left(\frac{\partial \mathbf{F}_{CL}}{\partial \dot{q}_2} + \mathbf{B}_L(.,2) + \frac{\partial \mathbf{C}}{\partial \dot{q}_2} \right) \right]_{2 \times 1} \quad \text{(A-39)}$$

Finally, the partial derivatives terms $\dfrac{\partial \mathbf{f}_1}{\partial \mathbf{u}}$, $\dfrac{\partial \mathbf{f}_2}{\partial \mathbf{u}}$, $\dfrac{\partial \mathbf{f}_3}{\partial \mathbf{u}}$,and $\dfrac{\partial \mathbf{f}_4}{\partial \mathbf{u}}$ of Equation (A-10) are given by

$$\frac{\partial \mathbf{f}_1}{\partial \mathbf{u}} = [0]_{2 \times 2}, \quad \frac{\partial \mathbf{f}_2}{\partial \mathbf{u}} = [\alpha 1]_{2 \times 2}, \quad \frac{\partial \mathbf{f}_3}{\partial \mathbf{u}} = [0]_{2 \times 2}, \quad \frac{\partial \mathbf{f}_4}{\partial \mathbf{u}} = \left[\alpha_2 \times \Lambda \times [\mathbf{I}_M]^{-1} \right]_{2 \times 2} \quad \text{(A-40)}$$

The matrix **G** of Equation (2-13) was derived using the procedure described in Section 2-2, and it has been found that its expression is given by:

$$\mathbf{G} = c\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{A-41}$$

VITA AUCTORIS

NAME	Frank Ciuca
PLACE OF BIRTH	Bucharest, Romania
YEAR OF BIRTH	1975
EDUCATION	Riverside Secondary School, Windsor, Ontario 1990-1992
	University of Windsor, Windsor, Ontario 1992-1996 B. Sc.
	University of Windsor, Windsor, Ontario 1996-1999 M. Sc.